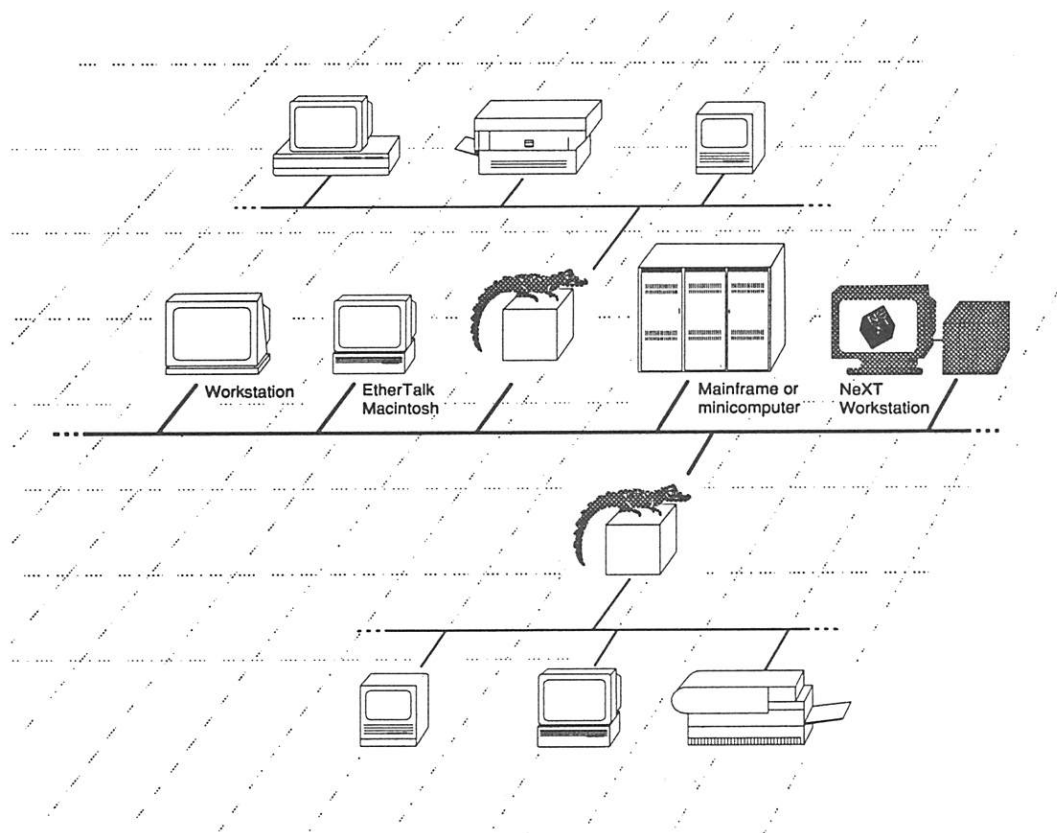


GatorBox[®]

Reference

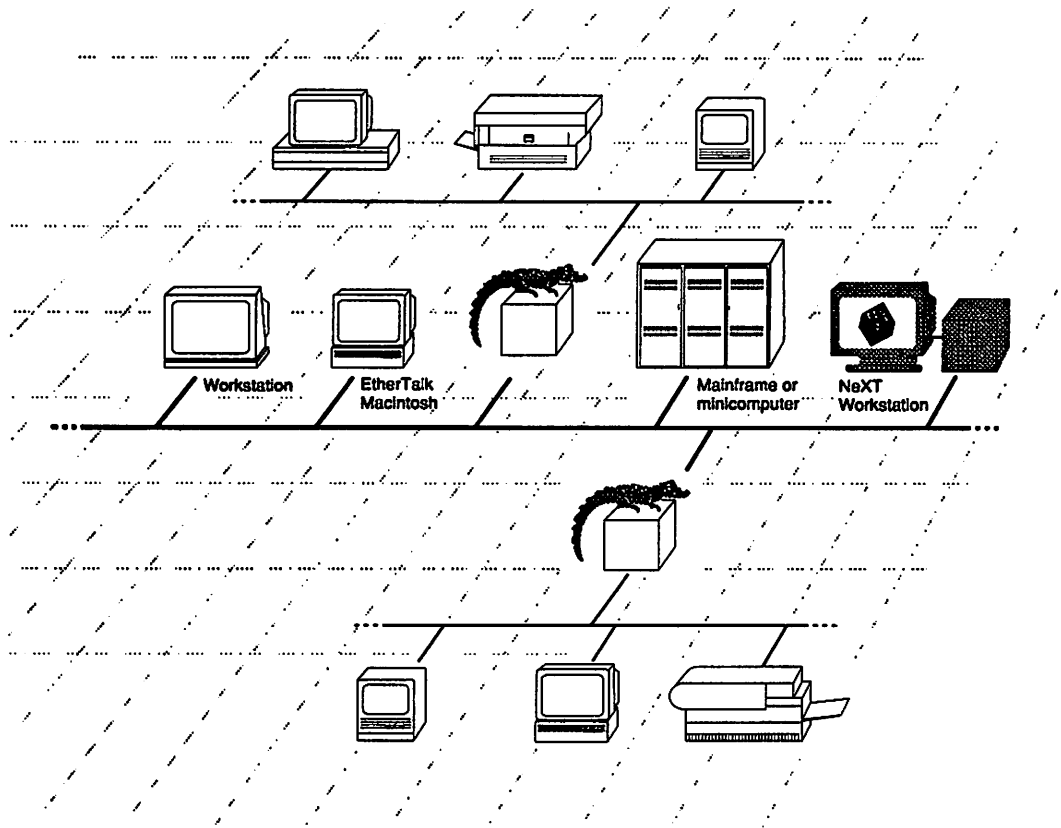
Release 2.0
August 1991



GatorBox[®]

Reference

Release 2.0
August 1991



This manual applies to release 2.0 of the GatorBox software (GatorSystem, GatorPrint, or GatorShare), which runs on the original GatorBox, and the GatorBox CS software (GatorSystem CS, GatorPrint CS, or GatorShare CS), which runs on the GatorBox CS, the GatorMIM CS, and the GatorBox CS/Rack. Updates to this manual will be distributed as document updates or new revisions.

Your comments about this manual are welcome. Use the forms at the back of the manual or address your comments to:

Technical Services
Cayman Systems
26 Landsdowne Street
Cambridge, MA 02139
Telephone: (617) 494-1999 (9:00 A.M. to 6:00 P.M. EST)
FAX: (617) 494-5167
internet support@cayman.com
AppleLink CAYMAN.TECH

APPLE COMPUTER, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY, OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

GatorBox, Cayman Systems, and the Cayman logo are registered trademarks of Cayman Systems, Inc. GatorKeeper, GatorMIM, GatorSystem, GatorPrint, and GatorShare are trademarks of Cayman Systems, Inc.

Apple, the Apple logo, AppleLink, AppleShare, the AppleShare icon, AppleTalk, A/UX, HyperCard, ImageWriter, LaserWriter, and Macintosh are registered trademarks of Apple Computer, Inc. Apple File Exchange, EtherTalk, Finder, and LocalTalk are trademarks of Apple Computer, Inc. The AppleShare icon, the trademark AppleShare, and AppleShare Workstation Software are exclusive property of Apple Computer, Inc. licensed to Cayman Systems, Inc.

AT&T and UNIX are registered trademarks of AT&T. Ethernet is a registered trademark of Xerox Corporation. FastPath is a trademark of Shiva Corporation. Microsoft and MS-DOS are registered trademarks of Microsoft Corporation. NCSA Telnet is a trademark of the Board of Trustees of the University of Illinois at Champaign-Urbana. NFS and Sun are trademarks of Sun Microsystems, Inc. PhoneNET is a trademark of Farallon Computing. MMAC is a trademark of Cabletron Systems, Inc. DECnet is a trademark of Digital Equipment Corporation.

Copyright © 1991 Cayman Systems.

All rights reserved.

Printed in the United States of America

Contents

How to Use This Manual

Who should use this manual	xi
Deciding what you should read	xi
What this manual covers	xii
Documentation conventions	xiii
For more information	xiv
Cayman Technical Services	xv

Chapter 1 — GatorBox Hardware and Software

What is the GatorBox?	1-1
What is the GatorBox software?	1-2
GatorSystem/GatorSystem CS	1-2
GatorPrint/GatorPrint CS	1-3
GatorShare/GatorShare CS	1-3
What is GatorInstaller?	1-4
What is GatorKeeper?	1-4
GatorDefaults file	1-4
GatorDatabase	1-4
GatorBoxes window elements	1-5
GatorKeeper menu bar	1-6
GatorKeeper menu commands	1-8
Apple menu	1-8
About GatorKeeper	1-8
Chooser	1-8
File menu	1-8
New	1-8
Open	1-9
Close	1-9
Save	1-9
Save As	1-9
Save Info as TEXT File	1-9
Page Setup	1-10
Print	1-10
Quit	1-10

Edit menu	1-11
Undo	1-11
Cut	1-11
Copy	1-11
Paste	1-11
Clear	1-12
Select All	1-12
Windows menu	1-12
GatorBoxes	1-12
Servers	1-13
TFTP Server Info	1-13
Special menu	1-14
Status	1-14
Diagnostics	1-15
Info	1-17
Statistics	1-18
Cleanup View	1-20
Rename GatorBox	1-20
Change Password	1-21
Restart GatorBoxes	1-22
Reload Software	1-22
View menu	1-23
by Icon	1-23
by Name	1-23
Lookup in Zone	1-24
Server Access menu	1-25
Apply Authentication	1-25
Add Server	1-26
Remove Server	1-26
Configuration Options window	1-27
GatorKeeper dialog boxes	1-28
TCP/IP dialog boxes	1-28
TCP/IP Configuration	1-28
MacIP Options	1-30
Additional TCP/IP MacIP Parameters	1-31
AppleTalk routing dialog boxes	1-33
AppleTalk Routing	1-33
AppleTalk Filter	1-36
KIP Options	1-38
AppleTalk Tunnel	1-40
Zone List	1-41

DECnet routing dialog boxes	1-42
DECnet Configuration.....	1-42
GatorPrint printing dialog boxes.....	1-43
Printer Configuration	1-43
File sharing dialog boxes	1-45
GatorShare Servers.....	1-45
AppleShare-to-NFS.....	1-45
NFS Mount Points	1-46
User/Group Info	1-48
Low Level Options.....	1-49

Chapter 2 — Network Basics

What is a network?	2-1
Network components	2-2
Nodes.....	2-2
Hosts	2-2
Servers and clients	2-2
Backbone network.....	2-3
Repeaters, bridges, routers, and gateways	2-3
Network media	2-4
Coaxial cable.....	2-4
Twisted pair cable.....	2-5
LocalTalk	2-5
LocalTalk connector boxes.....	2-5
Do's and Don'ts of LocalTalk.....	2-6
PhoneNET	2-6
PhoneNET connector boxes.....	2-6
Do's and Don'ts of PhoneNET	2-7
Thick Ethernet.....	2-7
Do's and Don'ts of thick Ethernet.....	2-8
Thin Ethernet	2-8
Do's and Don'ts of thin Ethernet	2-8
Twisted pair Ethernet	2-9
Do's and Don'ts of twisted pair Ethernet	2-9
Terminating a network	2-9
Network protocols	2-9
Network addressing.....	2-10
Binary, decimal, and hexadecimal numbers	2-11
Packets and datagrams	2-12

Chapter 3 — TCP/IP and MacIP

What is TCP/IP?	3-1
Internet Protocol (IP).....	3-1
Transmission Control Protocol (TCP)	3-1
User Datagram Protocol (UDP).....	3-2
File Transfer Protocol (FTP)	3-2
TELNET.....	3-2
Internet Control Message Protocol (ICMP).....	3-2
Simple Mail Transfer Protocol (SMTP).....	3-3
IP addressing.....	3-3
Hardware address	3-3
Internet (IP) address	3-3
Network segment.....	3-4
Node segment	3-4
IP address classes	3-5
Broadcasts and broadcast addresses.....	3-6
Address Resolution Protocol (ARP)	3-7
ARP cache	3-7
Proxy ARP	3-7
Reverse Address Resolution Protocol (RARP)	3-8
IP subnetting.....	3-8
Subnet mask.....	3-10
Broadcast addresses for subnet networks	3-12
Subnetting a LocalTalk network	3-13
IP routing	3-13
Routing tables	3-13
Routing Information Protocol (RIP).....	3-15
What is MacIP?	3-16
Dynamic MacIP address assignment.....	3-16
Static MacIP address assignment.....	3-17
MacIP on subnetted LocalTalk	3-18
AppleTalk Address Resolution Protocol (AARP).....	3-18
Restricting MacIP services to LocalTalk	3-19
NCSA Telnet	3-20
NCSA Telnet Settings File	3-21
NCSA Telnet config.tel file	3-21
config.tel syntax.....	3-22
Sample config.tel file	3-22

Chapter 4 — AppleTalk

What is AppleTalk?	4-1
Link Access Protocol (LAP)	4-1
Datagram Delivery Protocol (DDP)	4-1
Name Binding Protocol (NBP)	4-2
Routing Table Maintenance Protocol	4-2
AppleTalk zones	4-3
Routing tables	4-3
Zone Information Protocol (ZIP)	4-4
How AppleTalk works	4-5
Simple AppleTalk networks	4-5
Node addresses	4-5
Dynamic node address assignment	4-5
AppleTalk transmissions	4-6
NBP Lookups (NBPLkUp)	4-6
NBP Replies	4-7
Complex AppleTalk networks	4-8
NBP Broadcast Requests (NBPBrRq)	4-8
Seed, nonseed, and soft seed routers	4-10
Seed router	4-11
Nonseed router	4-11
Soft seed router	4-12
AppleTalk Phase 1/Phase 2	4-13
AppleTalk Phase 1	4-13
AppleTalk Phase 2	4-13
Phase 1/Phase 2 transition	4-14
AppleTalk tunnels	4-15
Network filtering	4-16
Device (NBP) filtering	4-17
Stay-in-zone filtering	4-18
Laser filtering	4-20
Device name (tilde) filtering	4-21
Kinetics Internet Protocol (KIP)	4-22
UDP port range	4-22
atalkad	4-23
atalkatab	4-23
/etc/atalk.local	4-24
Columbia AppleTalk Package (CAP)	4-24

Chapter 5 — DECnet

What is DECnet?	5-1
DECnet addresses	5-1
What is a DECnet router?	5-2
How does DECnet routing work?	5-2
Routing tables	5-3
Minimum cost calculation	5-4
Hello messages	5-5
Router messages	5-5

Chapter 6 — UNIX-to-LocalTalk Printing

About lpr	6-1
Operating systems that support lpr	6-2
About PAP	6-4
How GatorPrint works	6-5
Physical and logical printers	6-6
Connection attempts	6-7
Multiple printers and print queues	6-7
Printer name	6-8
Printer type	6-8
Compatible printers	6-9
PostScript translation	6-9
International character mapping	6-10
/etc/printcap file	6-12
Examples	6-14
Unique logical printers for each host	6-14
One UNIX host sends print jobs to the GatorBox	6-16
What NOT to do	6-16

Chapter 7 — NFS-to-AppleShare File Sharing

What is file sharing?	7-1
About Network File System (NFS)	7-2
Directories and pathnames	7-2
Mount points	7-3
/etc/exports file	7-4
Required NFS daemons	7-4
portmapper	7-4
nfsd	7-5
mountd	7-5
NFS security	7-5
/etc/hosts file	7-5

User access security	7-6
/etc/passwd file	7-6
/etc/group file	7-7
File access security	7-7
Network Information System (NIS)	7-8
How NIS works	7-9
PCNFSD	7-9
AppleShare	7-10
File formats	7-10
PC AppleShare	7-11
AppleShare security	7-12
AppleShare access privileges	7-12
GatorShare	7-13
.DESKTOP file	7-14
File creation times/dates	7-14
File name mapping	7-14
Byte-range locking	7-16
GatorShare security	7-17
User access matching	7-17

Chapter 8 — GatorBox Administration

TELNET shell	8-1
TELNET syntax	8-1
Sample TELNET commands	8-2
reset alap	8-2
reset enet	8-2
restart	8-2
show ip routes	8-2
show ip arp	8-3
show appletalk routes	8-3
show appletalk arp	8-4
show appletalk zones	8-4
show appletalk interfaces	8-4
show decnet nodes	8-5
show decnet circuits	8-5
show decnet status	8-6
show share	8-7
show alap	8-7
show enet	8-8
show crash	8-9
show memory	8-9

sho dump	8-10
status.....	8-10
Simple Network Management Protocol (SNMP)	8-10
Internet-standard MIB.....	8-11
Cayman's private MIB.....	8-12

Appendix A — Glossary

Appendix B — GatorBox SNMP MIB

Index

Reader Reply Card

How to Use This Manual

This manual is a general reference for the GatorBox hardware and software. It provides background information about TCP/IP and AppleTalk networking for new GatorBox users. Experienced GatorBox users can use the *GatorBox Reference* for answers to specific technical questions or for a detailed explanation of GatorBox functions.

Who should use this manual

This manual is intended for network administrators who are responsible for setting up and maintaining the GatorBox hardware and software.

Deciding what you should read

- ▶ **Hardware** — If you purchased a GatorBox CS, GatorMIM CS, or GatorBox CS/Rack, read the hardware manual that accompanied your gateway for information about:
 - ▷ Connecting your GatorBox to your Ethernet and LocalTalk networks
 - ▷ Using the GatorInstaller utility to load the GatorSystem CS, GatorPrint CS, or GatorShare CS software into the GatorBox CS flash EPROM memory
 - ▷ Running hardware diagnostics to test the GatorBox CS circuits and connections.

If you own an original GatorBox, read the *Setting Up Your GatorBox* manual, which accompanied your 2.0 software update, for information about connecting your GatorBox to your Ethernet and LocalTalk networks and downloading software and configuration settings from a Macintosh or TFTP server on your network.

- ▶ **Software** — Read the *GatorBox User's Guide* for information about running the GatorKeeper application to set up your GatorBox to support TCP/IP services, AppleTalk routing, DECnet routing, UNIX-to-LocalTalk printing, and AppleShare-to-NFS file sharing.
- ▶ **Troubleshooting** — Read the *GatorAid* Technical Services handbook for a discussion of questions frequently asked by Cayman customers.

What this manual covers

Here's what you will find in this manual:

- ▶ **Chapter 1**, "GatorBox Hardware and Software," provides an overview of the GatorBox hardware, the GatorSystem CS/GatorPrint CS/GatorShare CS software that runs on the GatorBox, and the GatorKeeper menus and dialog boxes.
- ▶ **Chapter 2**, "Network Basics," provides an introductory discussion of network concepts.
- ▶ **Chapter 3**, "TCP/IP and MacIP," describes the TCP/IP networking protocols and functions, including IP addressing, address resolution, and subnetting, and the MacIP implementation of the TCP/IP protocols on the Macintosh.
- ▶ **Chapter 4**, "AppleTalk," describes the AppleTalk networking protocols and functions, including AppleTalk routing, Phase 1/Phase 2 AppleTalk, and filtering.
- ▶ **Chapter 5**, "DECnet," describes concepts and background for implementing the DECnet routing functions.
- ▶ **Chapter 6**, "UNIX-to-LocalTalk Printing," describes concepts and background for implementing the printing functions in the GatorPrint CS and GatorShare CS software.
- ▶ **Chapter 7**, "NFS-to-AppleShare File Sharing," describes concepts and background for implementing the file sharing functions in the GatorShare CS software.

- ▶ **Chapter 8**, “GatorBox Administration,” describes the TELNET shell syntax and commands and presents an overview of Simple Network Management Protocol (SNMP) support in the GatorBox.
- ▶ **Appendix A**, “Glossary,” provides a brief definition of terms used in the GatorBox documentation set.
- ▶ **Appendix B**, “GatorBox SNMP MIB,” lists the private Management Information Base (MIB) used to query the GatorBox via SNMP about network statistics and traffic.

Documentation conventions

This manual uses certain presentation conventions to make the information easier to find and understand:

- ▶ **Menu commands and button names** appear in *italic sans serif* type face; for example:

The *About GatorKeeper* command lists the version number and copyright notice of your GatorKeeper software.

- ▶ **Computer text** appears in Courier (monospace) type face; for example:

The syntax for the group file is:

```
sales:*:14:jane,sandy,kim,peter,josiah
other:*:16:
marketing:*:17:andy,chris,michael,karen
qa:*:99:james,harry,ellen
```

- ▶ **User-entered text** appears in **Courier** (monospace) type face; for example:

To log in:

```
Username: Dublin
Password: *****
```

- ▶ **Menu commands that include ellipses (...)**, such as *Print...*, are presented without the ellipses in this manual.

- **Icons** identify special types of text:



- A **raised hand icon** indicates **important Caution** information. You should not proceed with an activity until you thoroughly read and understand Caution information.



- An **exclamation point icon** indicates **Alert** information. Alert messages provide important additional information about an activity or concept.



- A **talking head icon** indicates **Note** information. Notes provide additional or supplementary information about an activity or concept or suggestions for how to find out an item of information.



- A **book icon** indicates **Cross-Reference** information. Cross-references point to information in other manuals in your GatorBox documentation set that may be useful to understanding an activity or concept.



- A **GatorBox icon** indicates information applicable only to an original GatorBox.

- Unless otherwise noted, **the term “GatorBox”** applies to the GatorBox, GatorBox CS, GatorMIM CS, and GatorBox CS/Rack. When discussion in this manual applies only to the original GatorBox, a GatorBox icon will appear in the left margin.

For more information

You may want to read the following documents for more information about AppleTalk and TCP/IP networking topics:

- *AppleTalk Network System Overview*, Apple Computer, Addison Wesley © 1989.
- *Inside AppleTalk (2nd Edition)*, Sidhu, Andrews, and Oppenheimer, Addison Wesley © 1990.
- *Internetworking with TCP/IP (2nd Edition)*, Douglas Comer, Prentice-Hall © 1991.

- ▶ *NCSA Telnet for the Macintosh, Version 2.3*, University of Illinois at Urbana-Champaign, © 1989 (supplied, in compressed format, on the Network Applications disk that accompanies the GatorBox software).
- ▶ *The Simple Book*, Marshall T. Rose, Prentice Hall, © 1991.
- ▶ *Introduction to Administration of an Internet-based Local Network*, Charles Hedrick, Rutgers University, © 1988.

You can consult the Internet Request for Comments (RFCs) for background and technical information about TCP/IP protocols and standards. RFCs can be obtained by FTP from `nic.ddn.mil`.

Your UNIX system may also have on-line documentation (Man Pages) available. To use the UNIX Man Pages, type `man <topic>`.

Cayman Technical Services

Cayman's Technical Services staff is experienced in the installation and use of the GatorBox hardware and software. If this manual does not answer your questions about the GatorBox, you can call Cayman's Technical Services staff at (617) 494-1999 on all regular business days from 9:00 AM to 6:00 PM Eastern Time. You can also leave a message anytime by using one of the following:

FAX: (617) 494-5167

internet `support@cayman.com`

AppleLink CAYMAN.TECH

Chapter 1

GatorBox Hardware and Software

What is the GatorBox?

What is the GatorBox software?

What is GatorKeeper?

GatorKeeper menu bar

GatorKeeper menu commands

Configuration Options window

GatorKeeper dialog boxes



What is the GatorBox?

The GatorBox family of network gateway products connects a LocalTalk network with an Ethernet network. The GatorBox family includes:



- ▶ The **original GatorBox** is a desktop device in a metal housing. The original GatorBox comes with 1 MB of memory and requires an external power transformer. Instructions for connecting the GatorBox to your Ethernet and LocalTalk networks are provided in the *Setting Up Your GatorBox* manual.



- ▶ The **GatorBox CS** is a desktop device in a plastic housing. The GatorBox CS comes with 2 MB of memory and uses an internal power supply. Instructions for connecting the GatorBox CS to your Ethernet and LocalTalk networks are provided in the *Setting Up Your GatorBox CS* manual.



- ▶ The **GatorMIM CS** is a media interface module that fits in a Cabletron Multi-Media Access Center (MMAC™). The GatorMIM CS comes with 2 MB of memory. Instructions for installing the GatorMIM CS in a Cabletron Multi-Media Access Center (MMAC™) are provided in the *Setting Up Your GatorMIM CS* manual.



- ▶ The **GatorBox CS/Rack** is a rack mount version of the GatorBox CS that can be installed in a standard 19-inch device rack, mounted on a wiring closet wall, or stacked. Instructions for installing the GatorBox CS/Rack are provided in the *Setting Up Your GatorBox CS/Rack* manual.

Depending on the software that it runs, your GatorBox can perform several functions:

- ▶ **As an AppleTalk router**, the GatorBox lets Macintoshes on Ethernet communicate with devices on LocalTalk, such as other Macintoshes, LaserWriters, and AppleShare file servers.
- ▶ **As a TCP/IP gateway**, the GatorBox lets Macintoshes on LocalTalk networks access TCP/IP networks, making it possible for the Macintosh to function as a terminal connected to a UNIX host, exchange electronic mail with UNIX users, and transfer files to and from UNIX computers.

- ▶ **As a DECnet router**, the GatorBox lets Macintoshes on LocalTalk networks access DECnet networks, making it possible for the Macintosh to communicate with VAX, PDP, and other Digital computers.
- ▶ **As a print gateway**, the GatorBox lets UNIX computers send print jobs to printers on a LocalTalk network.
- ▶ **As a file-sharing gateway**, the GatorBox lets Macintoshes on LocalTalk or EtherTalk view and use NFS (Network File System) servers as AppleShare file servers.

What is the GatorBox software?

The software running in the GatorBox determines what services the GatorBox can provide. You can run one of three software images (GatorSystem, GatorPrint, or GatorShare) in your GatorBox. Table 1-1 illustrates the relationship between GatorSystem, GatorPrint, and GatorShare.

Image name	TCP/IP services	AppleTalk routing	DECnet routing	UNIX-to-LocalTalk printing	AppleShare-to-NFS file sharing
GatorShare	✓	✓	✓	✓	✓
GatorPrint	✓	✓	✓	✓	
GatorSystem	✓	✓	✓		

Table 1-1. GatorBox software

GatorSystem/GatorSystem CS

GatorSystem (GatorSystem CS, for the CS-class gateways) is the basic GatorBox operating software. When your GatorBox runs the GatorSystem software, it can function as a TCP/IP gateway to provide terminal services, mail gateways, and X-Windows capabilities between a Macintosh and a TCP/IP-based computer. The GatorBox lets a Macintosh running terminal emulation software, such as NCSA Telnet, issue commands to UNIX systems as if it were connected directly to the host computer. Other programs, such as X-Windows using MacTCP, also take advantage of this functionality.

Second, GatorSystem lets you use the GatorBox as an AppleTalk router to connect devices on a LocalTalk network to devices on an EtherTalk networks and to devices on remote LocalTalk networks. Using the GatorBox as an AppleTalk router lets Macintosh users on one LocalTalk or EtherTalk network communicate with Macintoshes and LocalTalk devices anywhere on the internet as though they were connected to the same LocalTalk cable.

Finally, GatorSystem lets you use the GatorBox as a DECnet router to connect devices on a LocalTalk network to devices on a DECnet network. Using the GatorBox as a DECnet router lets Macintosh users communicate with VAX users.



The GatorSystem CS software is installed in every GatorBox CS, GatorMIM CS, and GatorBox CS/Rack when it is manufactured. This software lets you use your GatorBox as a pre-configured for AppleTalk router as soon as you connect it to your LocalTalk and Ethernet networks.

GatorPrint/GatorPrint CS

GatorPrint (GatorPrint CS, for the CS-class gateways) is an upgrade to the GatorBox software that lets you send print jobs from UNIX computers using the lpr remote printer protocol to printers on your LocalTalk network. GatorPrint also provides the AppleTalk routing, TCP/IP terminal service, and DECnet routing functions of GatorSystem CS.

GatorShare/GatorShare CS

GatorShare (GatorShare CS, for the CS-class gateways) is an upgrade to the GatorBox software that lets you use the GatorBox as an AppleShare-to-NFS (Network File System) gateway. Macintosh users can then view computers supporting NFS as AppleShare file servers, giving them access to greatly expanded disk storage and file-sharing. GatorShare CS also provides the AppleTalk routing, terminal service, DECnet routing, UNIX-to-LocalTalk functions of GatorSystem and GatorPrint.

What is GatorInstaller?



GatorInstaller

The GatorInstaller utility loads the GatorBox operating software into the GatorBox CS, GatorMIM CS, or GatorBox CS/Rack flash EPROM memory. You do not need to run GatorInstaller unless you want to update the pre-installed GatorSystem CS software to a new version or unless you want to upgrade the software in your GatorBox to GatorPrint CS or GatorShare CS. Refer to the manual that accompanied your GatorBox hardware for information about running GatorInstaller.

What is GatorKeeper?



GatorKeeper

The GatorKeeper application configures your GatorBoxes for AppleTalk routing, TCP/IP services, UNIX-to-LocalTalk printing, and AppleShare-to-NFS translation. GatorKeeper lets you administer and monitor GatorBoxes on your LocalTalk network and in other AppleTalk zones on your internet. GatorKeeper lets you specify the addresses and protocols that each GatorBox will use to communicate with other hosts on your internet.

GatorDefaults file



GatorDefaults

The GatorDefaults file stores a complete set of configuration settings for a GatorBox. You create the GatorDefaults file the first time you run GatorKeeper. If you remove the GatorDefaults file from the folder containing the GatorKeeper application, GatorKeeper will create a new GatorDefaults file.

For information on using the GatorDefaults file to set up default configuration information, refer to the *GatorBox User's Guide*.

GatorDatabase



GatorDatabase

The GatorDatabase file contains information about each server that has been set up for a GatorBox running GatorShare CS. You must keep the GatorDatabase file in the same folder as the GatorKeeper application or in the System Folder.

GatorBoxes window elements

The GatorBoxes window (Figure 1-1) displays the icon of each GatorBox and GatorBox CS in the specified AppleTalk zone. You open the GatorBoxes window by selecting *GatorBoxes* from the Windows menu.

You use the icons in the GatorBoxes window to configure or monitor your GatorBoxes. For example, you obtain diagnostics information for a GatorBox by clicking its icon and choosing *Diagnostics* from the Special menu. To select more than one GatorBox, hold down the Shift key while you click the icon of each GatorBox.

Like other Macintosh applications, GatorKeeper includes a number of menus, windows, and dialog boxes. The remainder of this chapter describes:

- ▶ GatorKeeper menu bar
- ▶ GatorKeeper menu commands
- ▶ Configuration Options window
- ▶ GatorKeeper dialog boxes

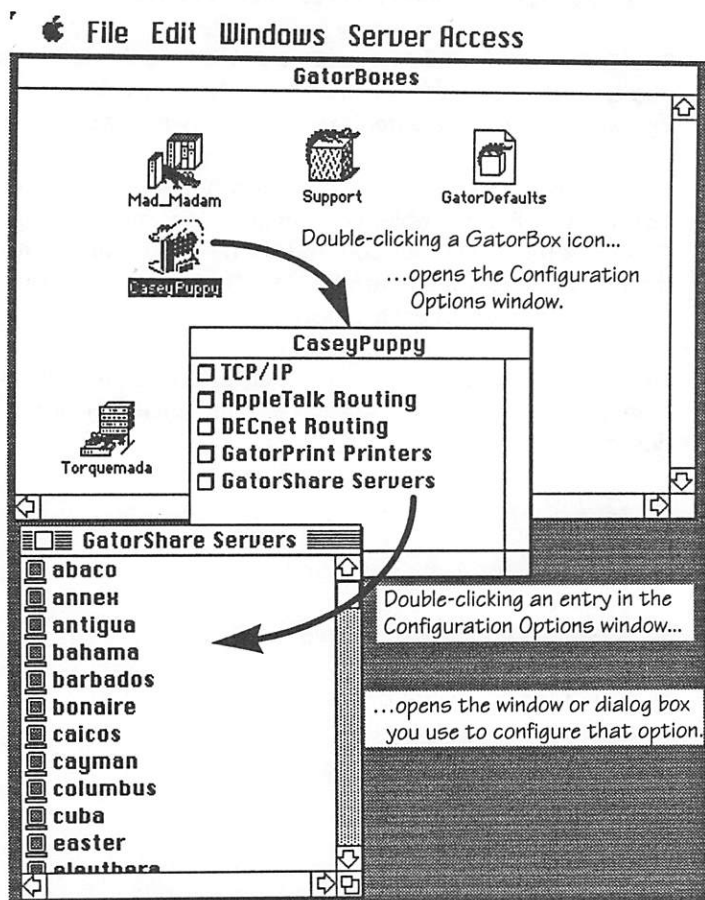


Figure 1-1. GatorKeeper windows and icons

GatorKeeper menu bar

The GatorKeeper menu bar at the top of the Macintosh screen displays the titles of the GatorKeeper pulldown menus. Four menus (Apple, File, Edit, and Windows) appear at all times when you are running GatorKeeper. The Special and View menus appear only when the GatorBoxes window is active. The Server Access menu appears only when the GatorShare Servers window is active.

The structure for the GatorKeeper menu bar is presented in Figure 1-2.

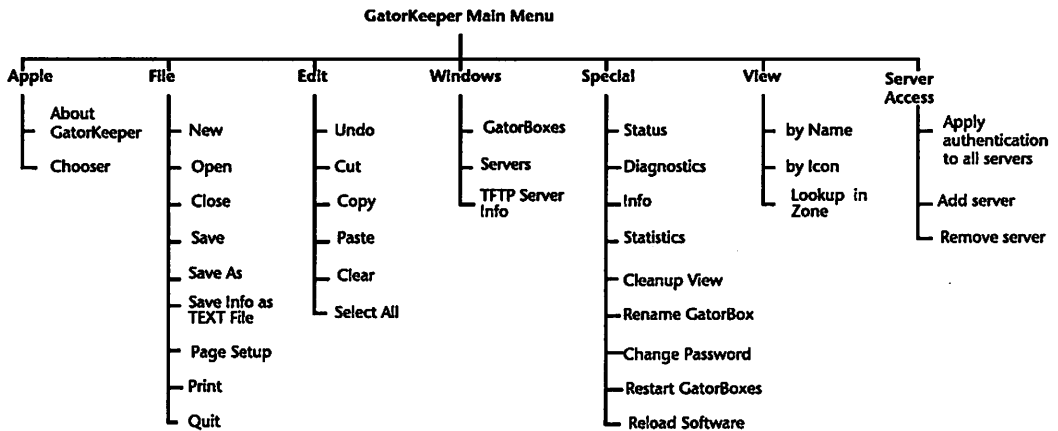


Figure 1-2. GatorKeeper menu bar

Many of the menu commands have a keyboard equivalent, allowing you to use the Command key in combination with another key as a shortcut for choosing a command from a menu. For example, holding down the Command (⌘) key and pressing **P** is the same as selecting **Print** from the File menu.

Some menu commands may be *dimmed* (displayed with gray letters instead of black letters in the menu). When a command is dimmed, you cannot use it or its keyboard equivalent. Dimmed commands often indicate that you need to supply more information for the command to be relevant. For example, you cannot select the **Status** command from GatorKeeper's Special menu until you specify a GatorBox by clicking its name or icon in the GatorBoxes window.

GatorKeeper menu commands

This section lists the commands on the GatorKeeper menu bar and briefly describes each one.

Apple menu

About GatorKeeper

Displays information about the software release level for the version of GatorKeeper you are running. Click inside the dialog box to remove it from the screen.

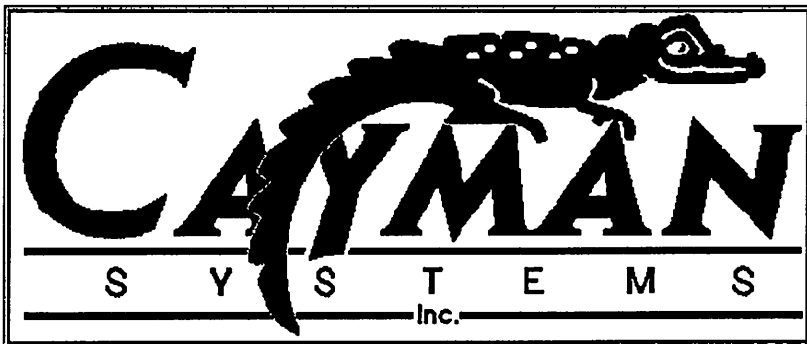


Figure 1-3. About GatorKeeper dialog box

Chooser

Displays the network resources, such as the printers or AppleShare file servers, in a specified zone.

File menu

New

⌘-N

If the GatorShare Servers window is open, selecting **New** opens the Add Servers dialog box (described on page 1-26).

Open

⌘-O

Opens the configuration file for a GatorBox.

Clicking the name or icon of a GatorBox in the GatorBoxes window and select **Open** from the File menu is identical to double-clicking the GatorBox name or icon.

Close

⌘-W

Closes the active window. If you make changes to a GatorBox configuration, you will be asked whether you want to save the changes before closing.

Choosing **Close** from the File menu is identical to clicking the close box in the window's title bar.

Save

⌘-S

Saves any changes made to a GatorKeeper configuration file:

- ▶ Changes to the configuration settings for an original GatorBox are saved to the GatorBox NVRAM and the *GatorBoxName* and GatorDatabase files.
- ▶ Changes to the configuration settings for a GatorBox CS are saved to the GatorBox flash EPROM memory. Saving changes to a GatorBox CS typically takes 10-20 seconds.

Save As

Not currently used.

Save Info as TEXT File

Saves information about a GatorBox to a text file. The information that is saved to a text file depends on which window or dialog box is active when it is selected:

- ▶ **Configuration information** — Clicking a GatorBox icon in the GatorBoxes window and choosing **Save Info as TEXT file** causes a full

description of the GatorBox configuration to be saved with the file name you specify.

- ▶ **Diagnostics window** — If the Diagnostics window for a GatorBox is active, choosing *Save Info as TEXT file* causes the GatorBox's diagnostics information to be saved with the file name you specify.
- ▶ **Info window** — If the Info window for a GatorBox is active, choosing *Save Info as TEXT file* causes the GatorBox's firmware and software release levels, serial number, network address, and crash information (if any) to be saved with the file name you specify.

Page Setup

Specifies settings for printing GatorKeeper information, such as paper size, page format, and reduction ratio.

Print

⌘-P

Prints information about a specified GatorBox to the AppleTalk printer selected in the Chooser. The information that is printed depends on which window or dialog box is active when the *Print* command is selected:

- ▶ **Configuration information** — Clicking a GatorBox icon in the GatorBoxes window and choosing *Print* causes a full description of the GatorBox configuration to be printed.
- ▶ **Diagnostics window** — If the Diagnostics window for a GatorBox is active, choosing *Print* causes the GatorBox's diagnostics information to be printed.
- ▶ **Info window** — If the Info window for a GatorBox is active, choosing *Print* causes the GatorBox's firmware and software release levels, serial number, network address, and crash information (if any) to be printed.

Quit

⌘-Q

Ends the GatorKeeper session and returns you to the Macintosh desktop or (with MultiFinder) a concurrently active application.

Edit menu

Undo

⌘-Z

Reverses the last action taken. *Undo* may not be available for some actions, such as saving information or assigning a password to a GatorBox.

Cut

⌘-X

Places a copy of selected text on the Clipboard and deletes the selected text from its current field. Cut text can then be pasted into other fields in GatorKeeper.

Copy

⌘-C

- ▶ When a GatorBox icon is selected in the GatorBoxes window, the *Copy* command copies the configuration settings from that GatorBox. You can then use the Paste command to copy those configuration settings to another GatorBox.
- ▶ When information in a text field is selected, the *Copy* command places a copy of selected text on the Clipboard but leaves the selected text unchanged. Copied text can then be pasted into other fields in GatorKeeper.

Paste

⌘-V

- ▶ When a GatorBox icon is selected in the GatorBoxes window, the *Paste* command copies the configuration settings (previously copied from another GatorBox with the *Copy* command) to the specified GatorBox.
- ▶ When information in a text field is selected, the *Paste* command copies text from the Clipboard to the specified field in a dialog box. If text is selected before the *Paste* command is issued, the text from the Clipboard replaces the selected text. You can use the *Paste* command to place the same information in multiple fields.

Clear

- ▶ When a GatorBox icon is selected in the GatorBoxes window, the *Clear* command erases the configuration settings from the GatorBox. If you select a GatorBox and choose *Clear*, GatorKeeper will prompt you to confirm that you want to erase its configuration settings.
- ▶ When information in a text field is selected, the *Clear* command erases the information but does not change the Clipboard.
- ▶ When crash information in the Info window is displayed, the *Clear* command erases the crash information from the GatorBox memory.

Select All



Selects all GatorBoxes in an AppleTalk zone. After you choose *Select All*, you can issue other commands, such as *Restart GatorBoxes*, to affect several GatorBoxes at the same time.

Windows menu

GatorBoxes

Opens the GatorBoxes window (Figure 1-4). By default, the GatorBoxes window displays the names or icons of each GatorBox in the GatorBox's own zone. You can specify another zone by using the Lookup in Zone command (described on page 1-24). You identify the GatorBox for which you want information or that you want to configure by clicking the device's name or icon in the GatorBoxes window.

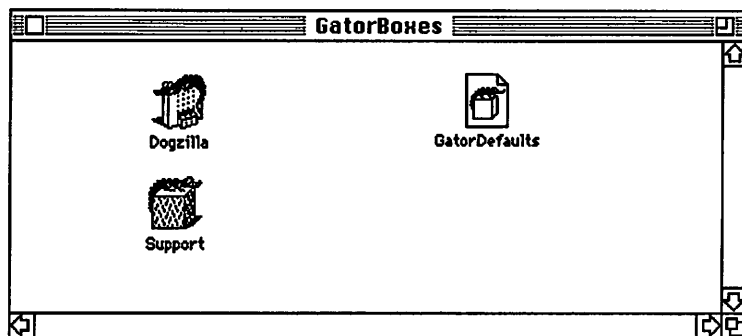


Figure 1-4. GatorBoxes window

Servers

Opens the Server List dialog box (Figure 1-5), which lets you maintain a list of the file servers that can be accessed from the GatorBox.

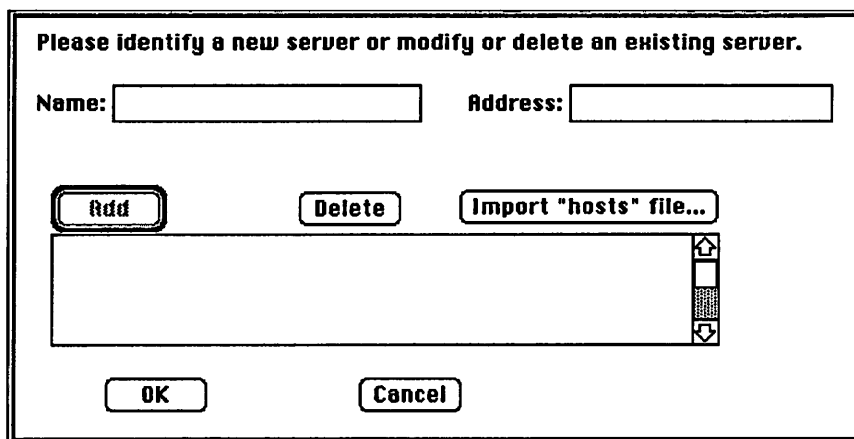


Figure 1-5. Servers List dialog box

TFTP Server Info



Provides download status information, such as the number of blocks transferred, when a GatorBox is downloading from a TFTP server.

Special menu

Status



Displays the operating status of one or more selected GatorBoxes in one of five columns:

- ▶ **Can't Find** — GatorBox is turned off or GatorKeeper is unable to locate the GatorBox on the network. *Can't Find* is the normal status for a GatorBox while it is restarting.
- ▶ **Unconfigured** — GatorBox has not been configured or its configuration settings have been erased.
- ▶ **Rebooting** — GatorBox is restarting.
- ▶ **Can't Download** — GatorBox is not able to download its software or its configuration information. Does not typically apply to GatorBox CS.
- ▶ **Running** — GatorBox is functioning properly.

Status				
Can't Find	Unconfigured	Rebooting	Can't download	Running
				 Dogzilla

Figure 1-6. Status window

Diagnostics

Displays the Diagnostic Messages window (Figure 1-7), which lists network activity messages and errors encountered by the selected GatorBox.

Diagnostic messages are retained when you close the Diagnostics Messages window. As long as GatorKeeper is running, you can close and open the Diagnostics Messages window without interrupting the diagnostics log.



Refer to the *GatorAid Technical Support Handbook* for a discussion of the standard diagnostic messages that appear when a GatorBox is restarted.

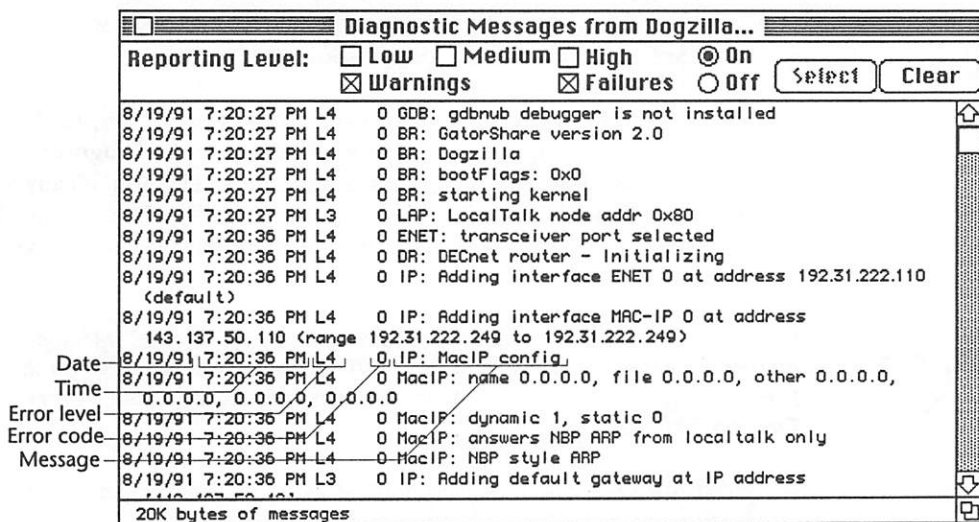


Figure 1-7. Diagnostics Messages window

Each diagnostic message consists of the following information:

- ▶ **Date** — The date, in *month:day:year* format, on which the GatorBox received the diagnostic message.
- ▶ **Time** — The time, in *hour:minute:second (hh:mm:ss:xx)* format, at which the GatorBox received the diagnostic message.
- ▶ **Reporting level** — The severity level of the diagnostic message:
 - ▷ **L1** (low-level informational) messages consist of trivial status messages generated by the GatorBox.

- ▷ **L2** (medium-level informational) messages consist of status messages that may help monitor network traffic.
- ▷ **L3** (high-level informational) messages consist of status messages that may be of interest to a user but that do not represent error conditions.
- ▷ **L4** (warning) messages describe recoverable error conditions and useful operator information.
- ▷ **L5** (failure) messages describe error conditions that may not be recoverable. You should contact Cayman Technical Services if you see L5 messages in the diagnostics log.

You can filter low-level messages from the diagnostics display by using the Error Reporting Level checkbox at the top of the Diagnostic Messages window. Selecting messages of one level automatically selects messages with higher level numbers. For example, clicking the *High* (level 3) check box causes the Diagnostics window to display messages with a level code of 3, 4, or 5.



Specifying that the Diagnostics log should record L1 (Low) or L2 (Medium) messages can reduce the performance of the GatorBox significantly. You should leave logging setting at High or Warning unless you are asked to do otherwise by Cayman Technical Support.

- ▶ **Error code** — The hexadecimal code identifying an error encountered by the GatorBox. Informational messages (L1–L3) usually have an error code of 0.
- ▶ **Message text** — A brief explanation of the message.

Info



Opens the Info window (Figure 1-8), which lists the serial number, Ethernet hardware address, software and firmware release levels, and network information for a GatorBox. You can display crash information for the GatorBox (if any) by clicking the icon in the upper right corner of the window.

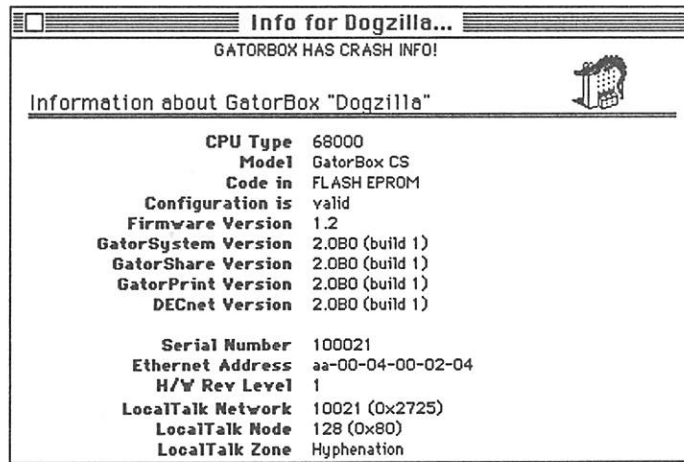


Figure 1-8. Info window

Clicking the GatorBox icon in the upper right corner of the Info window causes GatorKeeper to display crash statistics (if any) for the GatorBox (Figure 1-8).

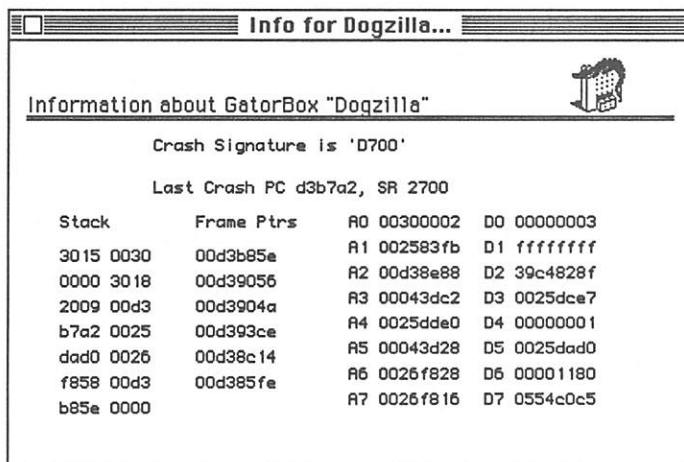


Figure 1-9. Info window — Crash statistics

Statistics

Displays memory use and load information (Figure 1-10) for a specified GatorBox:

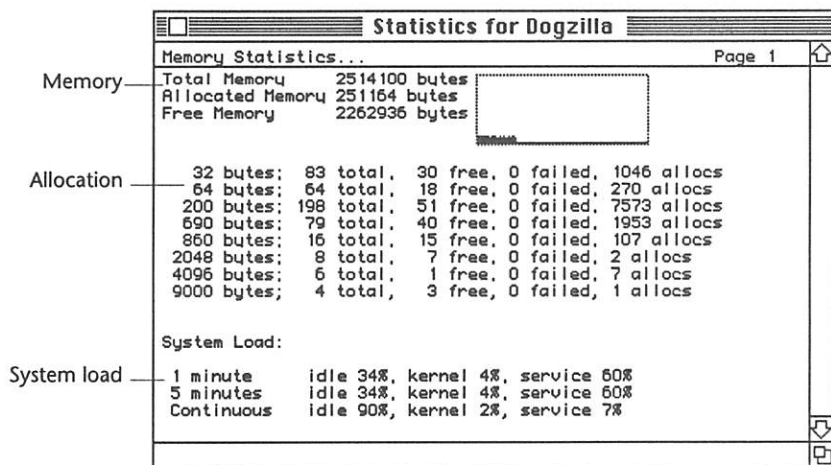


Figure 1-10. Statistics window

- ▶ **Memory** — The top area of the Statistics window summarizes overall memory usage in the GatorBox:
 - ▷ **Total memory** represents the memory initially available for allocation in the GatorBox after its GatorSystem CS, GatorPrint CS, or GatorShare CS software is loaded.
 - ▷ **Allocated memory** represents the amount of memory currently being used for processing by the GatorBox software.
 - ▷ **Free memory** represents the memory that has not yet been allocated.
 - ▷ The **bar graph** in the upper center of the dialog box summarizes memory usage over time.
- ▶ **Allocation** — The middle area of the Statistics window summarizes the use of memory partitions of various sizes in the GatorBox.
 - ▷ **# bytes** indicates the size of the memory partition.
 - ▷ **# total** indicates the total number of partitions of that size in the GatorBox.
 - ▷ **# free** indicates the number of partitions of that size that are not currently in use.
 - ▷ **# failed** indicates the number of failed allocation attempts for memory partitions of that size. Failed memory allocations are a normal part of the memory allocation process and are not typically grounds for concern.
 - ▷ **# allocs** indicates the total number of memory allocations for partitions of that size that the GatorBox has made since it was restarted.
- ▶ **System load** — Identifies running averages for CPU usage in the GatorBox:
 - ▷ **1 minute** indicates the running average in the last 60 seconds.
 - ▷ **5 minutes** indicates the running average in the last 5 minutes.

- ▷ **Continuous** indicates the running average since the GatorBox was restarted.

Cleanup View

Rearranges the icons in the GatorBoxes window neatly into rows and columns.

Rename GatorBox

Assigns a user-specified name to a GatorBox. A GatorBox comes configured with a name that includes its five- or six-digit serial number, such as GatorBox00019. This name can be seen in GatorKeeper after you turn on the GatorBox the first time.

You must restart a GatorBox after you assign it a new name for the name change to take effect.

If you rename a GatorBox, you must enter its new name and its IP address in the `/etc/hosts` file of each IP host you want to make accessible for file sharing.

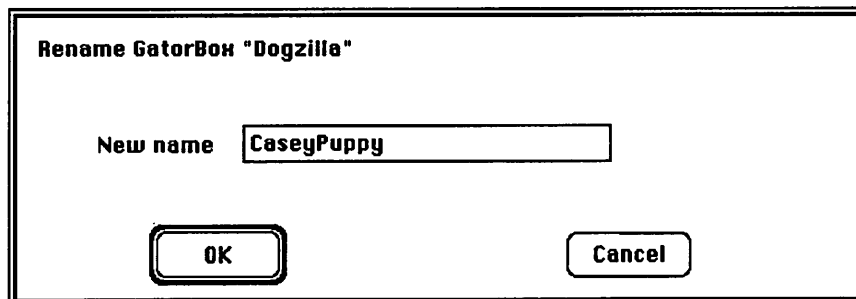
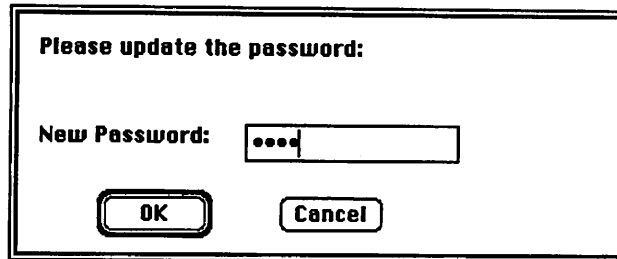


Figure 1-11. Rename GatorBox dialog box

Change Password

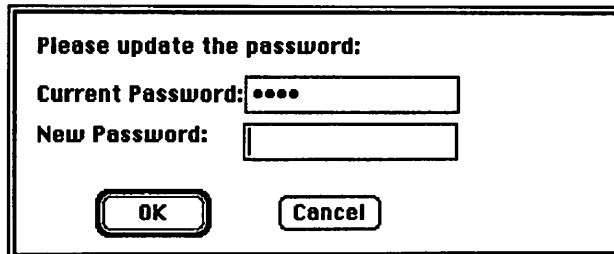
Assigns a password to a GatorBox or changes an existing password to prevent unauthorized modification of a GatorBox's configuration. If you have not previously assigned a password to the GatorBox, GatorKeeper displays a dialog box similar to the one shown in Figure 1-12.



A dialog box titled "Please update the password:". It contains a label "New Password:" followed by a text input field with four dots inside. Below the input field are two buttons: "OK" and "Cancel".

Figure 1-12. Using the Change Password dialog box to assign a new password

If you have previously assigned a password to the GatorBox, GatorKeeper displays a dialog box similar to the one shown in Figure 1-13.



A dialog box titled "Please update the password:". It contains two labels: "Current Password:" followed by a text input field with four dots, and "New Password:" followed by an empty text input field. Below the input fields are two buttons: "OK" and "Cancel".

Figure 1-13. Using the Change Password dialog box to change a password



The password you enter in the New Password field will become effective as soon as you click OK. You will not be prompted to re-enter the password. You do not have to save your changes or restart a GatorBox to assign it a password.

Restart GatorBoxes

Restarts one or more GatorBoxes whose icons are selected in the GatorBoxes window. When you select the **Restart GatorBoxes** command, you are asked to confirm the restart request.

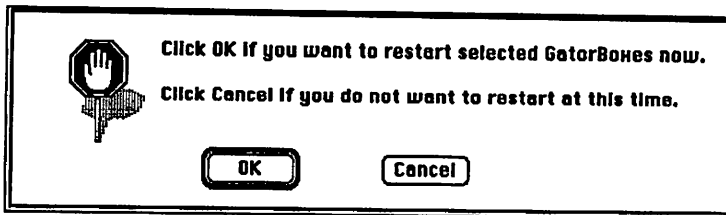


Figure 1-14. Restart GatorBox dialog box

- ▶ When an original GatorBox running GatorPrint or GatorShare restarts, it retrieves its configuration information from its designated download server.
- ▶ When a GatorBox CS restarts, it reads its configuration information from its internal configuration EPROM memory.

You must reset a GatorBox CS by using the **Restart GatorBoxes** command before any changes to its configuration settings take effect.

Reload Software



Reloads configuration settings in one or more original GatorBoxes whose icons are selected in the GatorBoxes window. When you select the **Reload Software** command, you are asked to confirm the restart request (Figure 1-15).

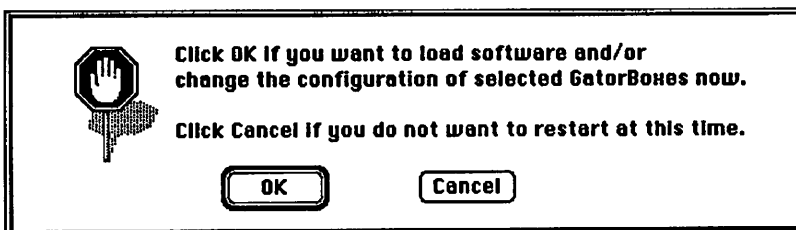


Figure 1-15. Reload Software dialog box

View menu

by Icon

Causes the GatorBoxes window to display the icon for the GatorDefaults file and the icon for each GatorBox in a selected AppleTalk zone.

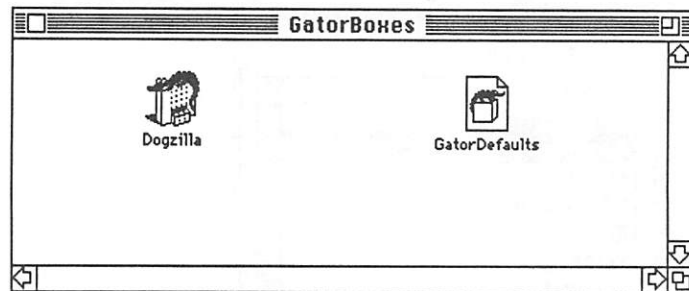


Figure 1-16. GatorBoxes window — View By Icon

by Name

Causes the GatorBoxes window to list each GatorBox in a selected AppleTalk zone alphabetically by name. When you select *By Name*, the GatorBoxes window also displays the status of each GatorBox in the list.

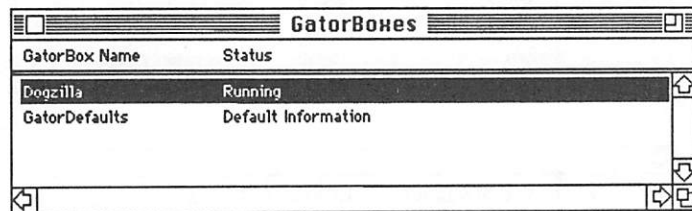


Figure 1-17. GatorBoxes window — View By Name

Lookup in Zone

Opens the Select Zone dialog box (Figure 1-18), which lets you specify the AppleTalk zone in which GatorKeeper will look for GatorBoxes. The GatorBoxes window title will indicate the name of the remote zone for which GatorBox icons are displayed (Figure 1-19).

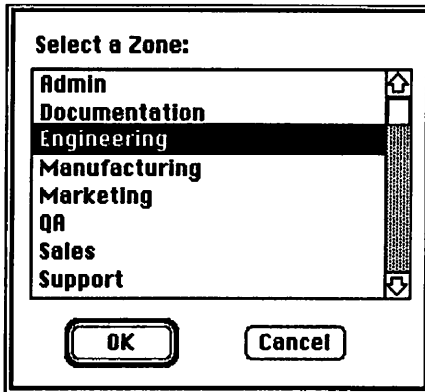


Figure 1-18. Select Zone dialog box

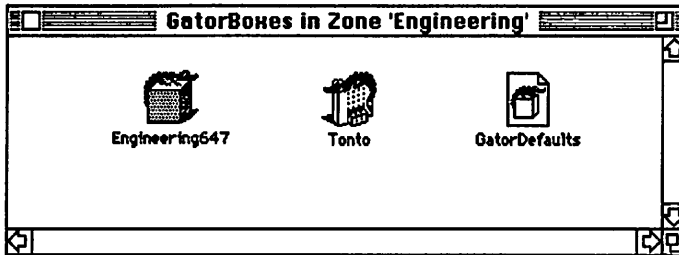


Figure 1-19. GatorBox icons from other zones



The lookup for GatorBoxes in a remote zone will typically take 10-20 seconds. GatorKeeper may add icons to the window as additional GatorBoxes make themselves known.

Server Access menu

Apply Authentication

Specifies the manner in which GatorShare CS users for all NFS servers in the Server List will be authenticated. Authentication specifications for specific servers can be set up with the User/Group Info dialog box (described on page 1-48).

Figure 1-20. Apply Authentication dialog box

- ▶ **Yellow Pages:** Radio button specifying that Yellow Pages (also called Network Information System, or NIS) will be used. If the *Yellow Pages* radio button is clicked, the *Domain Name* field appears.

- ▷ **Domain Name:** Name of the domain to which the volume belongs.



You can identify the domain to which a server volume belongs by logging in to the server and typing `domainname`.

- ▶ **Files:** Radio button specifying that the GatorBox will use password and group files to authenticate users. If the *Files* radio button is clicked, the *User file* and *Group file* fields appear.

- ▷ **User file:** The pathname to the passwd file for the server. The default value is `/etc/passwd`.

- ▷ **Group file:** The pathname to the group file for the server. The default value is `/etc/group`.
- ▶ **Use “pcnfsd” for user authentication:** Radio button specifying that the GatorBox will use PCNFS to authenticate users.

Add Server

Opens the Add Server dialog box (Figure 1-21), which lets you the servers in the GatorKeeper Server List that can be accessed from the selected GatorBox.

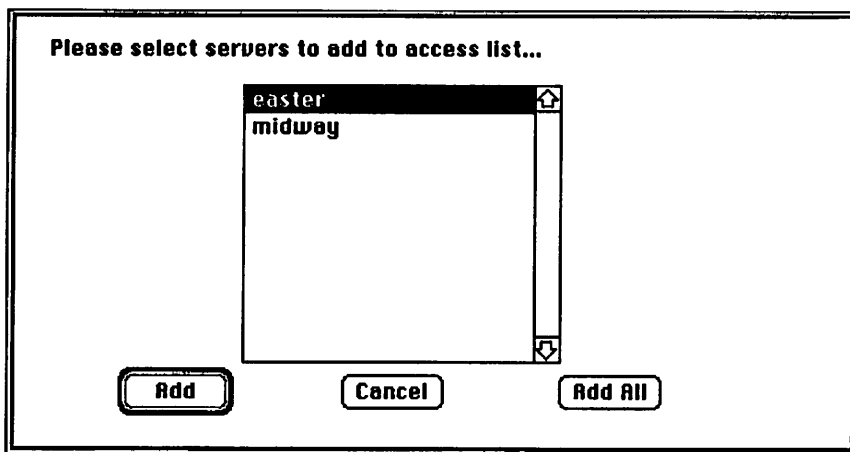


Figure 1-21. Add Server dialog box

Remove Server

Deletes one or more servers from the list of servers available through a GatorBox.

Configuration Options window

The Configuration Options window (Figure 1-22) lets you enter and modify the settings for your GatorBox. You access the Configuration Options window by double-clicking the name or icon of a GatorBox in the GatorBoxes window or by clicking the name or icon of a GatorBox and choosing the *Open* command from the File menu. The Configuration Options window displays the name of the GatorBox you selected as its title.

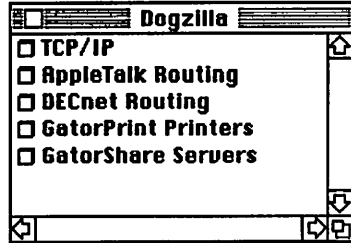


Figure 1-22. Configuration Options window

Depending on the version of GatorKeeper you are running, the Configuration Options window has as many as five selections:

- ▶ **TCP/IP** — Opens the TCP/IP Configuration dialog box (described on page 1-28).
- ▶ **AppleTalk Routing** — Opens the AppleTalk Routing dialog box (described on page 1-33).
- ▶ **DECnet Routing** — Opens the DECnet Routing dialog box (described on page 1-42).
- ▶ **GatorPrint Printers** — Opens the GatorPrint Configuration dialog box (described on page 1-43). The *GatorPrint Printers* option only appears when the GatorPrint or GatorShare software is installed in the GatorBox.
- ▶ **GatorShare Servers** — Opens the GatorShare Servers window (described on page 1-45). The *GatorShare Servers* option only appears when the GatorShare software is installed in the GatorBox.

GatorKeeper dialog boxes

This section presents each dialog box in GatorKeeper by related function and describes the fields in each dialog box. Refer to the *GatorBox User's Guide* for information about how to use the dialog boxes to configure your GatorBox.

TCP/IP dialog boxes

TCP/IP Configuration

Function: Sets up basic TCP/IP information, such as the IP address and subnet mask, for the GatorBox.

Access: Double-click the *TCP/IP* entry in the Configuration Options window.

Enter your TCP/IP parameters...

TCP/IP option IP address: 192.31.222.110
☒ On ☐ Off Broadcast Address: 192.31.222.0

☒ Default Gateway address ☒ Subnet mask
192.31.222.12 255.255.255.0

☒ Syslog host address User "MacIP" Options...
192.31.222.12

☒ Accept RIP packets
☒ Broadcast RIP packets

OK Cancel

Figure 1-23. TCP/IP Configuration dialog box

- ▶ **On/Off:** Radio buttons specifying whether TCP/IP services are turned on in the GatorBox.
- ▶ **IP Address:** The IP address assigned to the GatorBox. The IP address must not be assigned to any other device on your internet.
- ▶ **Broadcast Address:** The address the GatorBox uses to send broadcast messages to other hosts on the TCP/IP network. Refer to "Broadcasts

and broadcast addresses” on page 3-6 for more information about broadcast addresses.

- ▶ **Default Gateway address:** Checkbox specifying whether the GatorBox will direct messages to a default TCP/IP router if it does not know how to reach the destination host. A field for the IP address of the default TCP/IP router appears when you click the *Default Gateway* address checkbox.
- ▶ **Subnet mask:** Checkbox specifying whether a subnet mask is in use for the TCP/IP network. A field for the TCP/IP subnet mask appears when you click the *Subnet mask* checkbox.

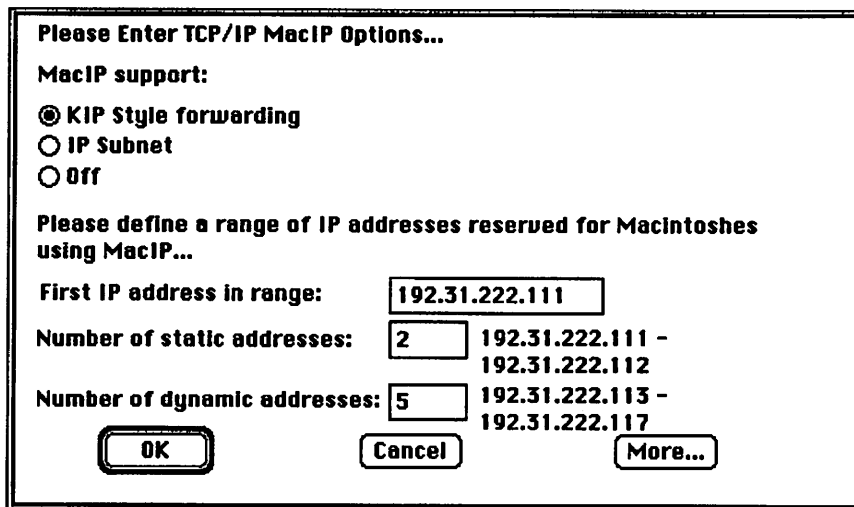
If you do not specify a subnet mask for the TCP/IP network, the GatorBox will use the default subnet mask for the network class. Refer to “Subnet mask” on page 3-10 for more information about subnetting and subnet masks.

- ▶ **Syslog host address:** Checkbox specifying whether the GatorBox should pass diagnostic messages to the syslog daemon on a specified host, which will log it to the appropriate system log, instead of to its own diagnostic message cache. A field for the IP address of the syslog host appears when you click the *Syslog host address* checkbox.
- ▶ **MacIP Options:** Button that opens the MacIP Options dialog box (described on page 1-30).
- ▶ **Accept RIP packets:** Checkbox specifying whether the GatorBox will listen to broadcasts from other IP routers about routes to other networks. For more information about RIP, refer to “Routing Information Protocol (RIP)” on page 3-15.
- ▶ **Broadcast RIP packets:** Checkbox specifying whether the GatorBox will broadcast information about its routing table to other IP routers on the internet.

MacIP Options

Function: Establishes settings for administration of MacIP addresses. Refer to "What is MacIP?" on page 3-16 for more information about MacIP.

Access: Click the *MacIP Options* button on the TCP/IP Configuration dialog box (described on page 1-29).



The dialog box is titled "Please Enter TCP/IP MacIP Options...". It contains the following elements:

- MacIP support:** Three radio buttons:
 - ☒ KIP Style forwarding
 - ☐ IP Subnet
 - ☐ Off
- Please define a range of IP addresses reserved for Macintoshes using MacIP...**
- First IP address in range:** A text field containing "192.31.222.111".
- Number of static addresses:** A text field containing "2", followed by the range "192.31.222.111 - 192.31.222.112".
- Number of dynamic addresses:** A text field containing "5", followed by the range "192.31.222.113 - 192.31.222.117".
- At the bottom are three buttons: "OK", "Cancel", and "More..."

Figure 1-24. MacIP Options dialog box

- ▶ **MacIP support:** Radio buttons specifying whether the GatorBox's LocalTalk network is part of the IP network or a separate IP subnet. The radio button selected determines which fields are displayed in the MacIP Options dialog box.
 - ▷ If the *KIP Style Forwarding* button is clicked, the LocalTalk network behind the GatorBox functions as part of the IP network.
 - ▷ If the *IP Subnet* button is clicked, the LocalTalk network behind the GatorBox functions as a separate IP subnet.
 - ▷ If the *Off* button is clicked, the LocalTalk network behind the GatorBox does not use MacIP functions.
- ▶ **LocalTalk IP Address:** (IP Subnet only) The IP address assigned to the LocalTalk subnet side of the GatorBox. The IP address entered in this

field and the subnet mask for the LocalTalk network (below) determine the network number for the LocalTalk subnet.

- ▶ **Subnet Mask:** (IP Subnet only) The mask indicating which bits of the IP address are part of the network number. Refer to "IP subnetting" on page 3-8 for more information about subnet masks.
- ▶ **First IP address in range:** The first IP address in the range reserved by the GatorBox for MacIP assignment. The default value is the IP address immediately after the GatorBox's own IP address.
- ▶ **Number of static addresses:** (KIP Style Forwarding only) The number of addresses that the GatorBox reserves for use by devices on its LocalTalk network. GatorKeeper displays the first and last IP addresses in the static MacIP address range. The sum of the numbers entered in this field and the *Number of dynamic addresses* field must be less than or equal to 64.
- ▶ **Number of dynamic addresses:** The number of IP addresses that the GatorBox reserves for dynamic assignment to devices on its LocalTalk network. GatorKeeper displays the first and last addresses in the address range for dynamic assignment. The sum of the numbers entered in this field and the *Number of static addresses* field must be less than or equal to 64.
- ▶ **More:** Button that opens the Additional TCP/IP MacIP Parameters dialog box (described on page 1-31).

Additional TCP/IP MacIP Parameters

Function: Specifies low-level TCP/IP settings.

Access: Click the *More* button on the MacIP Options dialog box (described on page 1-30).

Additional TCP/IP MacIP Parameters:

Name Server Address: 0.0.0.0

File Server Address: 0.0.0.0

User-Defined Addresses: 0.0.0.0
0.0.0.0
0.0.0.0
0.0.0.0

Select the style of AppleTalk ARP used: ☒ NBP (KIP) Style ARP
☐ DDP Style ARP

☐ Restrict MacIP service to LocalTalk

OK Cancel

Figure 1-25. Additional TCP/IP MacIP Parameters dialog box

- ▶ **Name Server Address:** The IP address of the host on the internet that acts as a name-to-IP address server.
- ▶ **File Server Address:** The IP address of the host on the internet that acts as a file server.
- ▶ **User-Defined Addresses:** Four fields for user-defined IP addresses.
- ▶ **AppleTalk ARP Style:** Radio buttons specifying the style of AppleTalk address resolution:
 - ▷ **NBP style ARP** uses NBP Lookups to resolve a network address. *NBP-style ARP* is the default setting and should be used unless your network requires compatibility with old-style network applications. **NBP-style ARP is the correct setting for using the GatorBox with GatorMail.**
 - ▷ **DDP Style ARP** uses DDP packets to resolve a network address. DDP-style ARP is supported for compatibility with applications written for older versions of the Kinetics FastPath. DDP-style ARP does not permit MacIP to assign IP addresses dynamically.

See “AppleTalk Address Resolution Protocol (ARP)” on page 3-18 for more information on NBP- and DDP-style ARP.

- **Restrict MacIP service to LocalTalk:** Checkbox specifying whether the GatorBox should restrict NBP Lookups to its LocalTalk network or whether it should forward NBP Lookups to EtherTalk and remote LocalTalk networks.

See “Restricting MacIP services to LocalTalk” on page 3-19 for more information on restricting NBP Lookups to LocalTalk.

AppleTalk routing dialog boxes

AppleTalk Routing

Function: Specifies basic configuration information for AppleTalk routing.

Access: Double-click *AppleTalk Routing* in the Configuration Options window.

Enter your AppleTalk Router parameters:

AppleTalk Routing: ☒ On ☐ Off

Filtering...
KIP Options...
AppleTalk Tunnels...

LocalTalk Network: **Soft Seed Port**
Number: 12362 Zone Name: Hyphenation

☒ Phase 1 EtherTalk: **Soft Seed Port**
Number: 2 Zone Name: Phase 1 Zone

☒ Phase 2 EtherTalk: **Soft Seed Port**
Network range: 22200 To: 22299 Zone List...

OK Cancel Defaults

Figure 1-26. AppleTalk Routing dialog box

- **On/Off:** Radio buttons specifying whether the GatorBox can route AppleTalk packets. **You cannot configure the GatorBox from a remote Macintosh if you turn off AppleTalk routing.**

- ▶ **Filtering:** Button that opens the AppleTalk Filtering dialog box (described on page 1-36).
- ▶ **KIP Options:** Button that opens the KIP Options dialog box (described on page 1-38).
- ▶ **AppleTalk Tunnels:** Button that opens the AppleTalk Tunnels dialog box (described on page 1-40).
- ▶ **(LocalTalk) Seed Port** — Popup menu specifying how the LocalTalk port will establish its network number and zone name:
 - ▷ **Seed Port** — Use the information entered in the *(LocalTalk Network) Number* and *(LocalTalk Network) Zone Name* fields to configure the LocalTalk network, regardless of what other routers may be advertising.
 - ▷ **Soft Seed Port** — Acquire network and zone information from another router on the LocalTalk network. If no other router provides the information, use the information entered in the *(LocalTalk Network) Number* and *(LocalTalk Network) Zone Name* fields to configure the LocalTalk network.
 - ▷ **Non Seed Port** — Acquire network and zone information from another router on the LocalTalk network. If no other router provides the information, turn off routing on the LocalTalk port.
- ▶ **(LocalTalk Network) Number:** The number assigned to the GatorBox's LocalTalk network. Each LocalTalk network must be assigned a unique network number. The default value for this field is a number based on the GatorBox serial number.
- ▶ **(LocalTalk Network) Zone Name:** Name of the AppleTalk zone that will be assigned to the GatorBox LocalTalk network. If a new zone name is entered, the LocalTalk network represents a new AppleTalk zone. If a zone name already in use is entered, the LocalTalk network belongs to that zone. The default value for this field is *LocalTalk<serial number>*.

- ▶ **Phase 1 EtherTalk:** Checkbox specifying whether the GatorBox supports Phase 1 AppleTalk on the Ethernet network. If the *Phase 1 EtherTalk* checkbox is clicked, the *(Phase 1) Seed Port* popup menu and the *Phase 1 EtherTalk Number* and *Zone Name* fields are displayed.
- ▶ **(Phase 1) Seed Port** — Popup menu specifying how the Phase 1 EtherTalk port will establish its network number and zone name:
 - ▷ **Seed Port** — Use the information entered in the *(Phase 1 Network) Number* and *(Phase 1 Network) Zone Name* fields to configure the Phase 1 EtherTalk network.
 - ▷ **Soft Seed Port** — Acquire network and zone information from another router on the Phase 1 EtherTalk network. If no other router provides the router configuration information, use the information entered in the *(Phase 1 Network) Number* and *(Phase 1 Network) Zone Name* fields to configure the Phase 1 network.
 - ▷ **Non Seed Port** — Acquire network and zone information from another router on the Phase 1 EtherTalk network. If no other router provides the information, turn off routing on the Phase 1 EtherTalk port.
- ▶ **(Phase 1 EtherTalk) Number:** Network number assigned to the EtherTalk network. All AppleTalk routers connected to the EtherTalk network must use the same network number to identify the network. The default value for this field is *1*.
- ▶ **(Phase 1 EtherTalk) Zone Name:** Zone name assigned to the Phase 1 EtherTalk network. All AppleTalk routers connected to the EtherTalk network must use the same zone name to identify the network. The default value for this field is *Phase 1 Zone*.
- ▶ **Phase 2 EtherTalk:** Checkbox specifying whether the GatorBox supports Phase 2 AppleTalk on the Ethernet network. If the *Phase 2 EtherTalk* checkbox is clicked, the *Phase 2 EtherTalk Network Range* fields are displayed.

- ▶ **(Phase 2) Seed Port** — Popup menu specifying how the Phase 2 EtherTalk port will establish its network number and zone name:
 - ▷ **Seed Port** — Use the information entered in the *(Phase 2 Network) Number* and *(Phase 2 Network) Zone Name* fields to configure the Phase 2 EtherTalk network.
 - ▷ **Soft Seed Port** — Acquire network and zone information from another router on the Phase 2 EtherTalk network. If no other router provides the router configuration information, use the information entered in the *(Phase 2 Network) Number* and *(Phase 2 Network) Zone Name* fields to configure the Phase 2 network.
 - ▷ **Non Seed Port** — Acquire network and zone information from another router on the Phase 2 EtherTalk network. If no other router provides the information, turn off routing on the Phase 2 EtherTalk port.
- ▶ **(Phase 2 EtherTalk) Network range:** First and last numbers in the range of network numbers assigned to the EtherTalk network. All AppleTalk routers connected to the EtherTalk network must use the same network number range to identify the network. If both the *Phase 1 EtherTalk* and the *Phase 2 EtherTalk* checkbox are selected, the beginning and ending numbers in the network range must be the same. The default values for these fields are **2 to 2**.
- ▶ **(Phase 2 EtherTalk) Zone List:** Button that opens the Zone List dialog box (described on page 1-41). The default value for the Phase 2 EtherTalk zone is *Phase 2 Zone*.

AppleTalk Filter

Functions: Sets up network filtering (for use with AppleTalk tunnels) and Name Binding Protocol filtering for the GatorBox.

Access: Click the *Filtering* button on the AppleTalk Routing dialog box (described on page 1-33).

Please designate AppleTalk networks to filter:

AppleTalk Network Number:

☒ Listen Only to These Remote Networks:

☐ Ignore These Remote Networks:

Please specify NBP Filtering Options:

☐ Stay-in-Zone Filter

☐ Laser Filter

☐ Tilde Filter

Figure 1-27. AppleTalk Filter dialog box

- ▶ **AppleTalk Network Number:** Number identifying the remote AppleTalk network that will be allowed or forbidden to exchange information via an AppleTalk tunnel.
- ▶ **Listen Only to These Remote Networks/Ignore These Remote Networks:** Radio buttons that specify whether the GatorBox should forward information about the remote AppleTalk networks specified in the scroll box.
 - ▷ If the **Listen Only to These Remote Networks** radio button is selected, the remote AppleTalk network must be explicitly identified by number before the GatorBox will pass information to and from it.
 - ▷ If the **Ignore These Remote Networks** radio button is selected, the GatorBox will pass information to and from any AppleTalk networks *except* those explicitly identified by number in the scroll box.
- ▶ **Stay-in-Zone Filter:** Checkbox specifying whether the GatorBox should filter NBP Lookup messages from devices on its LocalTalk network. Stay-in-zone filtering restricts users on the GatorBox LocalTalk network to their own zone, but lets users in other zones access devices

on the LocalTalk network. Refer to “Stay-in-zone filtering” on page 4-18 for more information about stay-in-zone filtering.

- ▶ **Laser Filter:** Checkbox specifying whether the GatorBox should filter NBP Reply messages from any device whose NBP type is LaserWriter. Laser filtering shields LaserWriters (and compatible printers) from users outside its zone. Refer to “Laser filtering” on page 4-20 for more information about laser filtering.
- ▶ **Tilde Filter:** Checkbox specifying whether the GatorBox should filter NBP Reply messages from any device whose name ends with a tilde (~). Tilde filtering shields a device whose name ends in a tilde from users outside its zone. Refer to “Device name (tilde) filtering” on page 4-21 for more information about device name (tilde) filtering.

KIP Options

Function: Configures the GatorBox to support the KIP protocols, which encapsulate AppleTalk packets inside UDP/IP packets. Refer to “Kinetics Internet Protocol (KIP)” on page 4-22 for more information about KIP.

Access: Click the *KIP Options* button on the AppleTalk Routing dialog box (described on page 1-33).

Please Enter AppleTalk KIP Options...

KIP Support (UDP Encapsulation):
☒ On ☐ Off

KIP AppleTalk Network Number: 8158

KIP AppleTalk Node Number: 110

KIP AppleTalk Zone Name: KIPzone

KIP IP Network Number: 192.31.222.109

☒ Configure Using "atalkad" ☒ Use New UDP Port Range (200)

"atalkad" Server Address:

Seed Port

OK Cancel

Figure 1-28. KIP Options dialog box

- ▶ **On/Off:** Radio buttons specifying whether the GatorBox supports the KIP protocols.
- ▶ **KIP AppleTalk Network Number:** The AppleTalk network number assigned to the KIP network. The KIP network number corresponds to the `mynet` and `bridgenet` fields in the `/etc/atalk.local` file on a CAP host. The KIP network number must be different than the numbers assigned to the EtherTalk and LocalTalk numbers for an internet.
- ▶ **(KIP) Seed Port** — Popup menu specifying how the KIP port will establish its network number and zone name:
 - ▷ **Seed Port** — Use the information entered in the *KIP AppleTalk Network Number* and *KIP AppleTalk Zone Name* fields to configure the KIP port.
 - ▷ **Non Seed Port** — Acquire network and zone information from another router on the network. If no other router provides the information, turn off routing on the KIP port.
- ▶ **KIP AppleTalk Node Number:** The node number for the GatorBox on the KIP network. The KIP node number corresponds to the `bridgenode` field in the `atalk.local` file on a CAP host.
- ▶ **KIP AppleTalk Zone Name:** The zone name assigned to the KIP network. The KIP zone name corresponds to the `myzone` fields in the `atalk.local` file on a CAP host. The KIP network number must be different than the numbers assigned to the EtherTalk and LocalTalk numbers for an internet.
- ▶ **KIP IP Network Number:** The number of the IP network on which the CAP host resides. The KIP IP network number does not correspond to any of the fields in the `atalk.local` file on a CAP host.
- ▶ **Configure Using “atalkad”:** Checkbox specifying whether the GatorBox should use information in the `/etc/atalk.local` and `/etc/atalkatab` files on a TCP/IP host to configure itself.
- ▶ **“atalkad” Server Address:** IP address of the server on the TCP/IP network on which the `/etc/atalk.local` and `/etc/atalkatab` files reside.

- **Use New UDP Port Range (200):** Checkbox specifying whether the GatorBox should use the current (200-206) or former (768-774) port number range to map the “well-known” DDP sockets.

AppleTalk Tunnel

Function: Sets up an AppleTalk tunnel between two GatorBoxes. For more information on AppleTalk tunnels, refer to “AppleTalk tunnels” on page 4-15.

Access: Click the *AppleTalk Tunnels* button on the AppleTalk Routing dialog box (described on page 1-33).

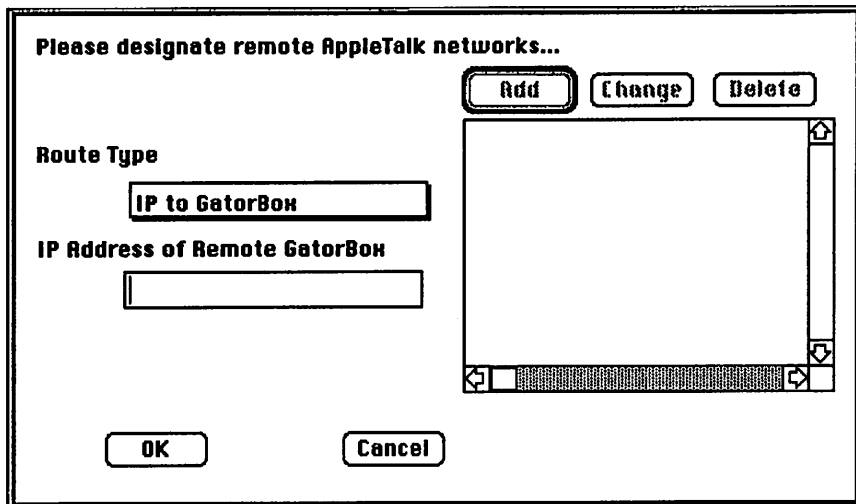


Figure 1-29. AppleTalk Tunnel dialog box

- **Route Type:** Specifies the type of AppleTalk tunnel you want to establish. At present, only *IP to GatorBox* is supported.
- **IP Address of Remote GatorBox:** IP address of the remote GatorBox that represents the other end of the AppleTalk tunnel.

Zone List

Function: Specifies the zone name or names associated with the Phase 2 EtherTalk network. At least one zone name must be set up for the Phase 2 EtherTalk network.

Access: Click the **Zone List** button on the AppleTalk Routing dialog box (described on page 1-33).

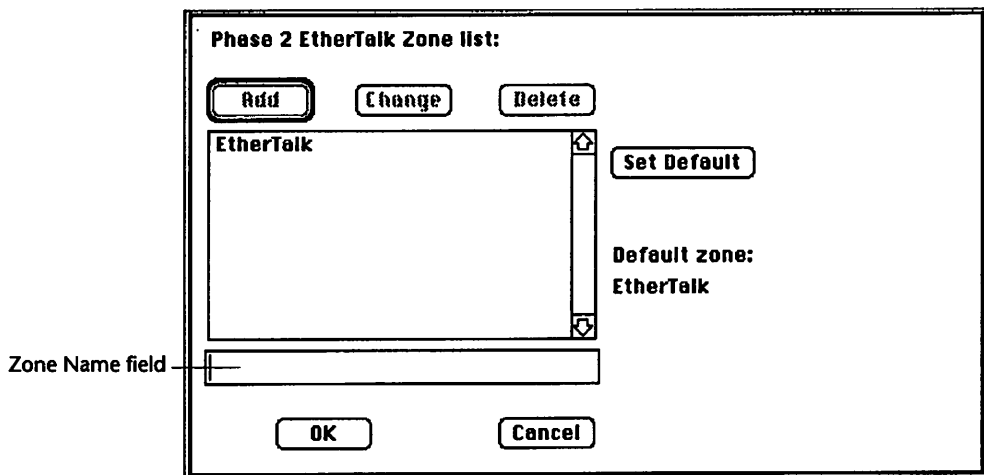


Figure 1-30. Zone List dialog box

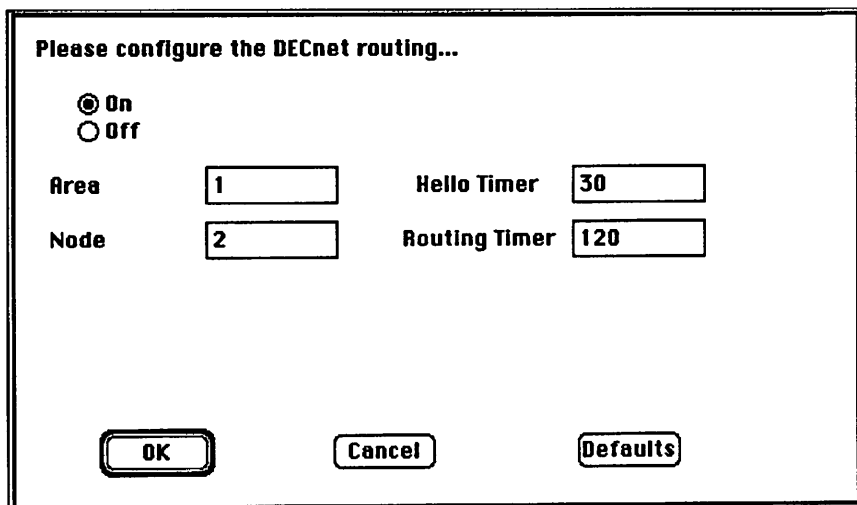
- ▶ **Zone Name:** The name of the AppleTalk zone to be added, modified, or deleted to the Phase 2 EtherTalk zone list.
- ▶ **Set Default:** Button that makes the zone name selected in the scroll box the default zone for the EtherTalk network.

DECnet routing dialog boxes

DECnet Configuration

Function: Configures the basic configuration settings for the GatorBox to function as a DECnet router.

Access: Click the *DECnet Routing* entry in the Configuration Options window.



The dialog box is titled "Please configure the DECnet routing...". It contains two radio buttons for "On" (selected) and "Off". Below these are four input fields: "Area" with the value "1", "Node" with the value "2", "Hello Timer" with the value "30", and "Routing Timer" with the value "120". At the bottom are three buttons: "OK", "Cancel", and "Defaults".

Figure 1-31. DECnet Configuration dialog box

- ▶ **On/Off:** Radio buttons specifying whether the GatorBox supports DECnet routing.
- ▶ **Area:** The number, in the range 1-63, identifying the DECnet area in which the GatorBox will function as a Level 1 router.
- ▶ **Node:** The number, in the range 1-1023, identifying the GatorBox's node address in the specified DECnet area.
- ▶ **Hello Timer:** The frequency, in seconds, with which the GatorBox will send Hello messages to end nodes in its area.
- ▶ **Routing Timer:** The frequency, in seconds, with which the GatorBox will send Router messages to other routers in its area.



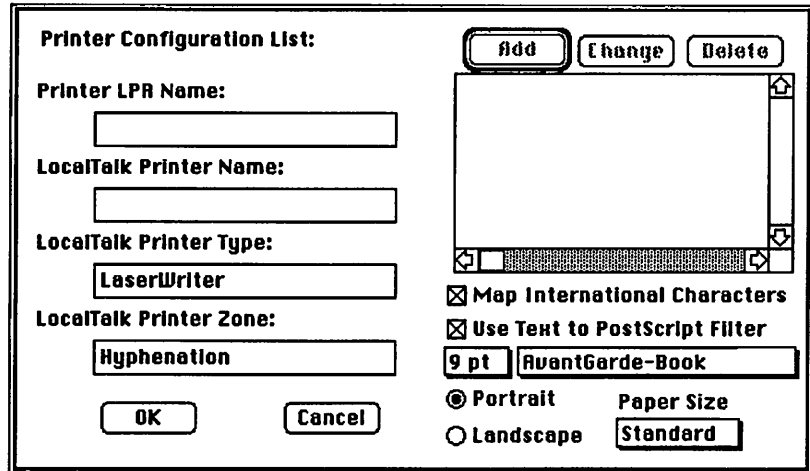
You should not change the timer values until you have discussed the proposed change with Cayman Technical Services.

GatorPrint printing dialog boxes

Printer Configuration

Function: Configures logical printer settings that let UNIX users submit print jobs to printers on the GatorBox LocalTalk network.

Access: Click the *GatorPrint Printers* entry in the Configuration Options window.



The dialog box is titled "Printer Configuration List:". It contains several input fields and controls:

- Printer LPR Name:** A text input field.
- LocalTalk Printer Name:** A text input field.
- LocalTalk Printer Type:** A dropdown menu with "LaserWriter" selected.
- LocalTalk Printer Zone:** A dropdown menu with "Hyphenation" selected.
- Buttons:** "Add", "Change", and "Delete" buttons are at the top right. "OK" and "Cancel" buttons are at the bottom left.
- Options:**
 - ☒ Map International Characters
 - ☒ Use Text to PostScript Filter
 - Font:** A dropdown menu showing "9 pt" and "AvantGarde-Book".
 - Paper Size:** Radio buttons for "Portrait" (selected) and "Landscape", and a dropdown menu showing "Standard".

Figure 1-32. Printer Configuration dialog box

- ▶ **Printer LPR Name:** The name entered in the `rp=` field of the `/etc/printcap` file, which identifies the remote printer to UNIX computers.
- ▶ **LocalTalk Printer Name:** The name the printer uses to register itself on the AppleTalk network, which is displayed in the Chooser when the printer's device type and zone are selected. Do not enter a space after the printer name.
- ▶ **LocalTalk Printer Type:** The Name Binding Protocol (NBP) device type for the printer. Apple printers use the following printer types:
 - ▷ **LaserWriter** — Used to identify most PostScript printers, such as an Apple LaserWriter or a Varityper laser printer.

- ▷ **ImageWriter** — Used to identify a QuickDraw printer, such as an ImageWriter I or II.

Some printer manufacturers may use their own NBP type codes to identify their printers. Consult your printer documentation for more information about the type code used by your printer.

- ▶ **LocalTalk Printer Zone:** The name of the AppleTalk zone to which the LocalTalk printer belongs.
- ▶ **Map International Characters:** Checkbox specifying whether the GatorBox should translate ISO 8859-1 hexadecimal codes to their international character equivalents.
- ▶ **Use Text to PostScript Filter:** Checkbox specifying whether the GatorBox should convert text files to PostScript before forwarding them to the LocalTalk printer. If the *Use Text to PostScript Filter* checkbox is turned on, the Printer Configuration dialog box displays several additional popup menus.
- ▶ **Type size:** Popup menu specifying the point size to which text will be converted. Files can be printed in 9, 10, 12, 14, or 18 point text.
- ▶ **Type face:** Popup menu specifying the type face to which text will be converted:
 - ▷ AvantGarde-Book
 - ▷ Bookman-Light
 - ▷ Courier
 - ▷ Helvetica
 - ▷ Helvetica-Narrow
 - ▷ *Helvetica-Oblique*
 - ▷ Palatino-Roman
 - ▷ Times-Roman
- ▶ **Portrait/Landscape:** Radio buttons specifying whether text files converted to PostScript will be printed in portrait (tall) or landscape (wide) format.
- ▶ **Paper size:** Popup menu specifying the paper size for which the converted text file will be formatted. The following options are available:

- ▷ **Standard** — 8.5" x 11" (21.8 cm x 28.2 cm)
- ▷ **U.S. Letter** — 8.5" x 11" (21.8 cm x 28.2 cm)
- ▷ **A4** — 8.2" x 11.6" (21 cm x 29.7 cm)
- ▷ **B5** — 7.2" x 10.1" (18.4 cm x 25.9 cm)

File sharing dialog boxes

GatorShare Servers

Function: Lists the NFS servers that have been set up for the selected GatorBox.

Access: Double-click *GatorShare Servers* in the Configuration Options window.

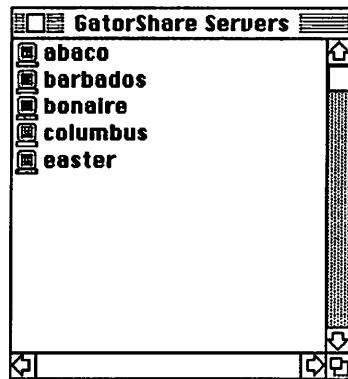


Figure 1-33. GatorShare Servers window

AppleShare-to-NFS

Function: Specifies the name of an NFS volume set up as an AppleShare volume. It also provides access to the dialog boxes needed to set up GatorShare mount points and user authentication.

Access: Double-click an NFS server entry in the GatorShare Servers window (described above).

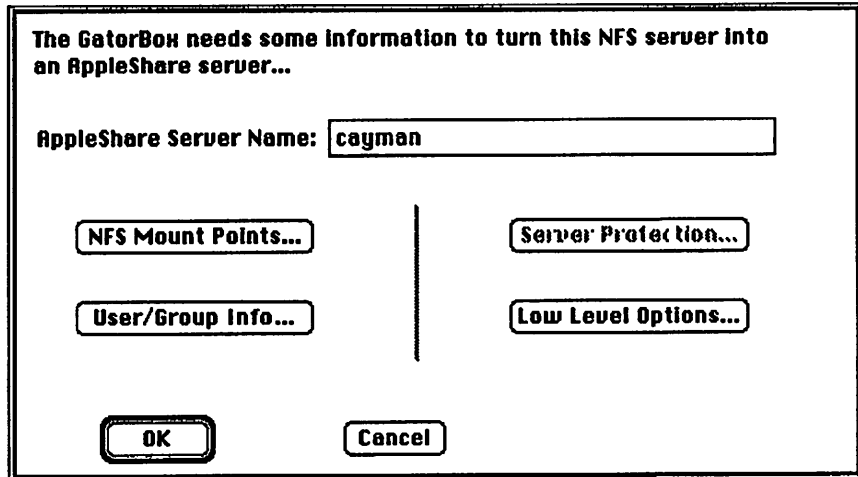


Figure 1-34. AppleShare-to-NFS dialog box

- ▶ **AppleShare Server Name:** Specifies the name that appears in the Chooser to identify the AppleShare volume on the NFS server. The AppleShare server name can be as many as 32 characters long. The server name can include spaces or special characters but cannot include a colon (:).
- ▶ **NFS Mount Points:** Button that opens the NFS Mount Points dialog box (described on page 1-46).
- ▶ **User/Group Info:** Button that opens the User/Group Info dialog box (described on page 1-48).
- ▶ **Low Level Options:** Button that opens the Low Level Options dialog box (described on page 1-49).

NFS Mount Points

Function: Sets up basic configuration information for each mount point on an NFS server.

Access: Click the *NFS Mount Points* button on the AppleShare-to-NFS window (described on page 1-45).

Figure 1-35. NFS Mount Points dialog box

- ▶ **NFS Mount Point:** The name of the mount point in UNIX directory syntax.
- ▶ **AppleShare Volume Name:** The name that will identify the mount point on the Macintosh desktop when it is mounted as an AppleShare volume.
- ▶ **NFS Path to .DESKTOP file:** The pathname to the .DESKTOP file for the mount point. The default value for this field is .DESKTOP.
- ▶ **Volume Password** — The password assigned to the NFS mount point to restrict access by GatorBox users linking to it as AppleShare volume. If the field is blank, the NFS mount point does not have a password.
- ▶ **AppleSingle/AppleDouble** — Radio buttons that specify whether files on the NFS mount point will be stored in AppleSingle or AppleDouble format. Refer to “File formats” on page 7-11 for more information about AppleSingle and AppleDouble file formats.



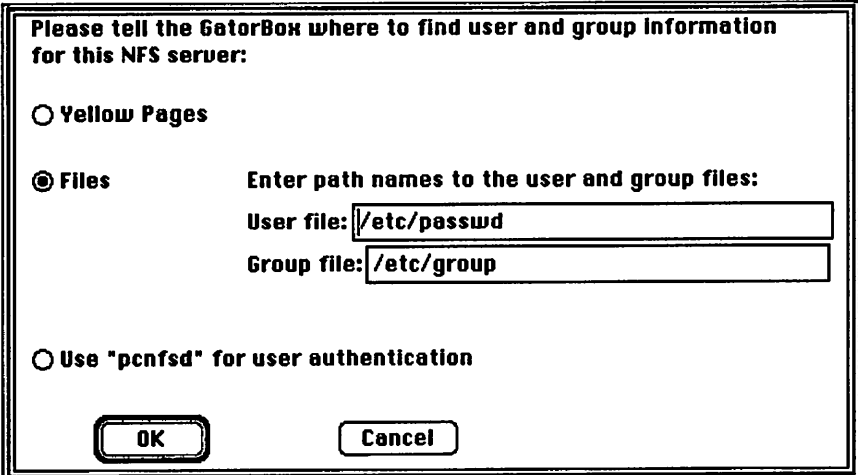
AppleSingle should only be used for archiving Macintosh files on NFS servers. Although you can copy a file to and from an AppleSingle volume with GatorShare, you should never create or modify a Macintosh file while it resides on an AppleSingle volume.

- ▶ **Show Volume(s):** Checkbox specifying whether the GatorBox can access the mount point specified in the scroll box.
- ▶ **Show invisible files:** Checkbox specifying whether GatorShare CS will include files with names beginning with a period when it lists the files on the mount point.
- ▶ **Convert TEXT files:** Checkbox specifying whether GatorShare CS converts the ASCII carriage return character in Macintosh applications to the ASCII line-feed character used by most UNIX applications.

User/Group Info

Function: Specifies the manner in which the GatorBox should authenticate a GatorShare CS user before allowing access to NFS volumes.

Access: Click the *User/Group* button in the AppleShare-to-NFS window (described on page 1-45).



The dialog box has a title bar and a main content area. The title bar contains the text: "Please tell the GatorBox where to find user and group information for this NFS server:". The main content area contains three radio buttons: "Yellow Pages", "Files", and "Use 'pcnfsd' for user authentication". The "Files" radio button is selected. To the right of the "Files" radio button is the text "Enter path names to the user and group files:". Below this text are two text input fields. The first field is labeled "User file:" and contains the text "/etc/passwd". The second field is labeled "Group file:" and contains the text "/etc/group". At the bottom of the dialog box are two buttons: "OK" and "Cancel".

Figure 1-36. User/Group Information dialog box

- ▶ **Yellow Pages:** Radio button specifying that Yellow Pages (also called Network Information System, or NIS) will be used. If the *Yellow Pages* radio button is clicked, the *Domain Name* field appears.
- ▶ **Domain Name:** Name of the domain to which the volume belongs.

- **Files:** Radio button specifying that the GatorBox will use password and group files to authenticate users. If the **Files** radio button is clicked, the **User file** and **Group file** fields appear.
- **User file:** The pathname to the passwd file for the server. The default value is `/etc/passwd`.
- **Group file:** The pathname to the group file for the server. The default value is `/etc/group`.
- **Use "pcnfsd" for user authentication:** Radio button specifying that the GatorBox will use PCNFSd to authenticate users.

Low Level Options

Function: Establishes fundamental configuration settings for exchanging files with NFS hosts.

Access: Click the Low Level Options button on the AppleShare-to-NFS dialog box (described on page 1-45).

NFS Low Level Options: ☐ Checksum Packets

Retry count: 10

Timeout (ms): 1000

Read Size (bytes): 4624

Write Size (bytes): 4624

Filename Mapping:

☒ No filename mapping ☐ 7 bit filenames Delimiter: :

☐ 8 bit filenames ☐ 7 bit alphanumeric filenames

Prefix used for AppleDouble resource files: %

OK Cancel Defaults

Figure 1-37. Low Level Options dialog box

- **Checksum packets:** Checkbox specifying whether the GatorBox performs checksum calculations on the UDP/IP packets it sends and

receives. Turning on the checksum function enhances packet integrity but slows transmission.

- ▶ **Retry count:** The number of times the GatorBox will resend unanswered packets to the NFS server. The retry count can be any number in the range 0 to 99. The default value is 10.
- ▶ **Timeout (ms):** The time, in milliseconds, that the GatorBox will wait for a response from the NFS server before issuing a retry. The timeout interval can be any number in the range 0-9999. The default value is 1000.
- ▶ **Read size:** The number of bytes in read requests the GatorBox sends to NFS servers. The read size can be any number in the range 0-4624. The default value is 4624. A smaller read size, such as 1024, is required if your server is a Sun-1, Sun-2, or Macintosh running A/UX version 1.0.
- ▶ **Write size:** The number of bytes in write requests the GatorBox sends to NFS servers. The write size can be any number in the range 0-4624. The default value is 4624. A smaller write size, such as 1024, is required if your server is a Sun-1, Sun-2, or Macintosh running A/UX version 1.0.
- ▶ **File name mapping:** Radio buttons specifying how the GatorBox should map illegal characters in Macintosh file and directory names. Mapped characters are replaced with a three-character string consisting of a delimiter character (specified in the *Delimiter* field) and the two-character hexadecimal value of the illegal character. File mapping options consist of:
 - ▷ **No filename mapping** — Do not convert filename characters.
 - ▷ **8 bit filenames** — Store any 8-bit character except slash (0x2F), null (0x00), or the delimiter character.
 - ▷ **7 bit filenames** — Store any 7-bit character except slash (0x2F), null (0x00), or the delimiter character. Replace extended 8-bit characters (0x80-0xFF), slash, null, and the delimiter.
 - ▷ **7 bit alphanumeric filenames** — Store any alphanumeric (0-9, a-z, A-Z) character, the underscore character, and the last period in the file name. Replace all other characters in Macintosh file names.

Figure 1-38 lists the hexadecimal equivalents for 7- and 8-bit characters.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
7-bit characters	00															
	10															
	20		!	"	#	\$	%	&	'	(	)	*	+	,	-	.
	30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>
	40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^
	60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n
	70	p	q	r	s	t	u	v	w	x	y	z	{		}	~
8-bit characters	80	Ä	Å	Ç	É	Ñ	Ö	Ü	á	à	â	ä	ã	å	ç	é
	90	ê	ë	í	ì	î	ï	ñ	ó	ò	ô	ö	õ	ú	ù	û
	A0	†	°	€	£	\$	•	¶	ß	®	©	™	'	¨	≠	Æ
	B0	∞	±	≤	≥	¥	μ	∂	Σ	Π	π	∫	ª	º	Ω	æ
	C0	¿	¡	¬	√	ƒ	≈	Δ	«	»	...		À	Ã	Ö	œ
	D0	-	—	"	"	'	'	+	◊	ÿ	ÿ	/	◻	<	>	fi
	E0	‡	·	,	„	%	Â	Ê	Á	Ë	È	Í	Î	Ì	Ó	Ô
	F0		Ò	Ú	Û	Ü	ı	ˆ	˜	˘	˙	˚	¸	˝	˞	˜

Figure 1-38. Character mapping table

- **Delimiter:** Character used to precede the hexadecimal number of illegal characters in file names. The delimiter character cannot itself be included in a file name.
- **Prefix used for AppleDouble resource file:** The character(s) the GatorBox will add to the beginning of a Macintosh file name to identify the file's resource fork when it stores the file on an NFS server. The default value is %.



Chapter 2

Network Basics

What is a network?

Network components

Network media

Network protocols

Network addressing

Packets and datagrams

C

C

C

What is a network?

Networking is a tool for communication. Networks let computer users communicate with one another, gather and exchange information, and share printing and file storage resources.

A *local area network* (LAN) connects devices in a limited area, such as a department in an office building or factory. A LAN is not usually defined in terms of number of users — a LAN can connect a few computers to a printer, or it can connect hundreds of computers and network devices.

A *wide area network* (WAN) connects devices in different locations into a single network. A WAN might link the computers in a corporation's main office to the computers in its branch offices. Where a LAN uses a dedicated network medium (cable) to transfer data, a WAN typically uses telephone lines or satellite links to transfer information.

An *internet* is the connection of two or more networks such that devices and users on one network can communicate with devices and users on the other networks. Because small networks are often easier to set up and maintain than large networks, an internet can simplify network administration. However, because each network in an internet retains its own network number, internets add a layer of complexity to communication between devices.

The benefits of connecting computer users by means of a network include:

- ▶ **Access to information** — Networks let users access information stored in a central location, such as a corporate database.
- ▶ **Information sharing** — Networks let users exchange files electronically, making it possible for users at different locations to work together on projects and documents.
- ▶ **Centralized file service** — Networks let users store files in central locations (file servers), making it easier to back up files regularly and to maintain consistent distribution of files and applications.
- ▶ **Improved communication with other users** — Networks let users send electronic mail messages, files, and documents to each other.

- ▶ **Better use of network resources** — Networks let users share expensive computer resources, such as printers, modems, or disks, making the resources cost-effective.

Network components

A network typically includes many types of components, including computer hosts, printers, or servers, and connection components, such as network software, cables, and routing devices.

Nodes

Any device on a network that has a unique address is called a *node*. A node can be a computer (such as a personal computer, minicomputer, or mainframe), a network resource (such as a printer or file server), or a network device (such as a router or gateway).

Not every device on a network is a node. For example, a repeater or modem does not usually need a network address to function. Consequently, devices such as these are not considered nodes.

Hosts

Hosts are computers on a network that act as a central processing unit for end users. A host can be a personal computer used by one person or a mainframe computer with hundreds of terminal connections.

Servers and clients

A host that provides services for other devices on the network is called a *server*.

- ▶ **File servers** store files and folders that other computers on the network can use.
- ▶ **Print servers** process print requests submitted by client computers, freeing the client computer to do other tasks.
- ▶ **Name servers** map host names to IP addresses, simplifying network administration.
- ▶ **Mail servers** store messages sent from one mail client to another.

A *client* is a host (or a process on a host) that uses the services provided by a network server. Clients submit a request to a server and wait for a response.

A machine on the network can be both a client and a server. For example, the host that acts as a mail server for an entire network may obtain user authorization information from a Yellow Pages server.

Backbone network

Many internets use a high-speed network as a *backbone* to connect smaller networks. An internet backbone functions like an interstate highway: it moves large volumes of traffic from one connection point to another quickly and eases cross-network traffic congestion. Using a backbone to connect networks lets the network administrator isolate individual networks without disrupting traffic on other networks.

Repeaters, bridges, routers, and gateways

Before you can connect two networks together, you need to add a device that will pass packets from one side to the other. The type of device you need to add depends on the types of networks you are connecting and the functions that you want the device to perform



- ▶ A **repeater** amplifies a network signal traveling from one network segment to another. When two network segments are connected by a repeater, the segments share the same network number (and, in the case of AppleTalk networks, the same zone name).



- ▶ A **bridge** passes packets from one network to another network of the same type. Unlike a repeater, which forwards electrical signals, a bridge forwards packets from one network to another. Bridges differ from routers in that they use physical (hardware) addresses instead of logical (network) addresses.



- ▶ A **router** scans packets from one network, determines if they are intended for another network of the same type, and forwards (*routes*) them from one network to the other when appropriate. Routers let you expand your internet beyond the size of a single network and improve network performance by isolating local traffic on each connected network.

Each router on an internet maintains a **routing table**, which lets it determine how to send a packet to a remote network. Routing tables list other networks on the internet and the routers to which packets should be directed to reach them. A routing table can also list the distance (in number of routers, or “hops”) and efficiency (in terms of “cost”) between this router and other networks. When a router has more than one path for directing packets to a remote network, the routing table enables the router to determine the most efficient route for sending the packets.



- A **gateway** passes packets from one type of network to another. Unlike routers, gateways typically perform some sort of protocol translation. For example, a GatorBox running GatorShare translates AppleShare file requests to NFS file requests.

Network media

Network media refers to the physical conductors (cabling) for a network. Coaxial and twisted pair cable are the most common designs for network media. A network may use one medium instead of another because it offers a better mix of characteristics. For example, thick (coaxial) Ethernet cabling offers relatively high transmission speeds, *noise resistance* (ability to keep signals free of distortion from outside interference), and *bandwidth* (capacity), but it is more expensive to install than other network media.

Coaxial cable

Coaxial cable consists of a central wire, around which is an insulating layer, a woven wiring layer, and an external shielded insulation layer. Coaxial cable is moderately expensive, but it offers good transmission speeds and resistance to electrical interference.

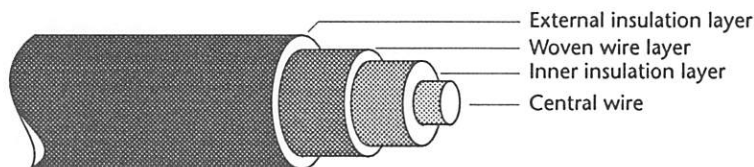


Figure 2-1. Coaxial cable cross-section

Twisted pair cable

Twisted pair cable consists of two insulated wires twisted around each other inside a plastic or rubber casing. Twisted pair cable is relatively inexpensive and easy to install, but it offers lower noise resistance than coaxial cable.

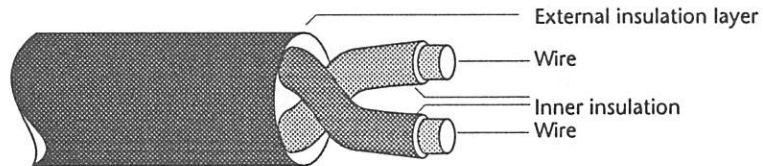


Figure 2-2. Twisted pair cable cross-section

Twisted pair cabling can be either *unshielded* or *shielded*. Unshielded twisted pair cabling is subject to electrical interference from other devices in the area. Shielded twisted pair cabling uses a third wire wrapped around the two network wires to improve noise resistance.

LocalTalk

LocalTalk is a shielded twisted pair cable designed by Apple. A LocalTalk network consists of a series of LocalTalk connector boxes connected by LocalTalk cables, and cable extenders. LocalTalk can support as many as 32 nodes over a 1,000-foot cable run. LocalTalk allows transmission speeds up to 230.4 kilobits per second.

A transceiver for sending and receiving information over LocalTalk networks is built into every Macintosh and LaserWriter. Setting up a simple LocalTalk network is usually a matter of making the physical connections between devices.

LocalTalk connector boxes

In addition to the cables through which packets travel, you need LocalTalk connector boxes for each device on your LocalTalk network. The LocalTalk connector box consists of a small box with two 3-pin miniature DIN ports on one end and a short cable and mini-circular 8-pin connector on the other end.

The LocalTalk connector box is self-terminating, so you do not need to terminate the ends of a LocalTalk network. The use of LocalTalk connector

boxes lets users remove devices from a LocalTalk network (by disconnecting them from the connector box) without disturbing traffic on the network.

Do's and Don'ts of LocalTalk

- ▶ You must connect a LocalTalk connector box to each device on your LocalTalk network.
- ▶ When you remove a device from a LocalTalk network, disconnect the connector box from the device rather than disconnecting the network from the connector box.
- ▶ Do not loop a LocalTalk network back on itself to form a circle configuration.
- ▶ You must turn off a device before you connect it to your LocalTalk network. A device that remains on when it is connected to the network will not verify that its node number is not already being used.

PhoneNET

Like Apple's LocalTalk cabling system, Farallon's PhoneNET lets you connect devices to an AppleTalk network. Unlike LocalTalk, however, PhoneNET uses ordinary telephone (unshielded twisted pair) cabling instead of a proprietary cable design as the basis of the network.

A PhoneNET network can support as many as 32 nodes over a 1,000-foot cable run. PhoneNET allows transmission speeds up to 230.4 kilobits per second.

PhoneNET connector boxes

You need PhoneNET connector boxes for each device on a PhoneNET network. A PhoneNET connector box is similar in appearance to a LocalTalk connector box: it consists of a small box with two telephone-type ports on one end and a short cord and DIN-8 plug on the other end.

Unlike LocalTalk connector boxes, PhoneNET connector boxes are not self-terminating. The PhoneNET terminator consists of a 100-ohm resistor mounted on an RJ11 (telephone-type) connector. You must include a

terminator on the connector boxes for the first and last devices on a PhoneNET network.

Do's and Don'ts of PhoneNET

- ▶ You must attach a PhoneNET connector box to each device on your PhoneNET network.
- ▶ If the first and last connector boxes on a PhoneNET network are not self-terminating, you must terminate the network by inserting a PhoneNET terminator in the unused port on the connector box.
- ▶ Do not loop a PhoneNET network back on itself to form a circle configuration.
- ▶ You must turn off a device before you connect it to your PhoneNET network. A device that remains on when it is connected to the network will not verify that its node number is not already being used.

Thick Ethernet

Thick Ethernet cable (also called standard Ethernet or 10Base5 Ethernet) is a coaxial cable about 0.5 inches in diameter. A thick Ethernet cable network can support up to 200 nodes on a single segment, which can be up to 1600 feet in length. By using repeaters to connect cable segments, a thick Ethernet network can support over 1000 nodes and be up to 8000 feet long. Thick Ethernet allows transmission speeds of up to 10 megabits per second.

Thick Ethernet requires an external transceiver to connect a device to the network. An Ethernet transceiver tap inserts small pins into holes in the cable's external insulation layer. These pins connect the woven metal layer and the central wire of the Ethernet cable. The transceiver is responsible for electrically isolating the node from the network.

Thick Ethernet networks require a terminator at either end. The terminator, which puts a resistor between the woven wire layer and the center wire, prevents reflection of electrical signals.

Do's and Don'ts of thick Ethernet

- ▶ You must terminate a thick Ethernet network at both ends to avoid signal reflection.
- ▶ If your GatorBox is connected to a multiport transceiver, you must verify that the transceiver's SQE heartbeat is turned off.

Thin Ethernet

Thin Ethernet (also called Cheapernet or 10Base2 Ethernet) is a less-expensive version of the thick Ethernet coaxial cable design. Although thin Ethernet is similar in appearance to the coaxial cable used to install cable television, its electrical characteristics are different, and one should not be substituted for the other. A thin Ethernet cable network can support up to 30 nodes on a single segment, which can be up to 640 feet in length. By using repeaters to connect cable segments, a thin Ethernet network can support over 1000 nodes and be up to 3200 feet long. Thin Ethernet allows transmission speeds of up to 10 megabits per second.

Unlike thick Ethernet, thin Ethernet does not require an external transceiver to connect a device to a network. Instead, hardware manufacturers frequently integrate Ethernet hardware and a BNC port into their products. BNC connectors make it easier to connect devices to the network cable.

Do's and Don'ts of thin Ethernet

- ▶ You must terminate a thin Ethernet network at both ends to avoid signal reflection.
- ▶ You must use a T-connector for each device on your thin Ethernet network. Never plug the thin Ethernet cable directly into a device's BNC port.
- ▶ You must attach the T-connector directly to the BNC port on a network device. Never add a cable segment between the T-connector and the device.



Twisted pair Ethernet

Twisted pair Ethernet (also called 10BaseT Ethernet) uses a pair of copper wires instead of a coaxial cable to transmit network signals. Twisted pair Ethernet is usually cheaper and easier to install than thin or thick Ethernet, but it may not offer as much protection from interference. Like coaxial Ethernet, twisted pair Ethernet allows transmission speeds of up to 10 megabits per second.

Do's and Don'ts of twisted pair Ethernet

- ▶ The GatorBox does not offer a twisted-pair Ethernet port. You must use a twisted-pair-to-thick-Ethernet or twisted-pair-to-thin-Ethernet transceiver.
- ▶ Unshielded twisted pair networks should be isolated from electrical devices, such as motors.

Terminating a network

Failing to terminate a network is a common source of network problems. If a network is not terminated properly, the electrical signals that carry data along the cable are reflected back through the cable when they reach the end of the network segment. Reflected signals introduce "noise" onto the network and degrade network performance. Network terminators at the ends of a network segment absorb these signals.

Network protocols

Protocols are the rules governing how devices communicate on a network. Protocols specify how devices locate one another, how they format and address messages, and how they share the network media for message transmission.

Network protocols typically involve three types of services:

- ▶ **Application services** let an application program on one computer communicate with a similar program on another computer.
- ▶ **Transport services** manage addressing and transmission control tasks.

- ▶ **Connection services** govern the transmission of a packet from one computer to another over the network.

The protocol that a network uses has little to do with the type of cable used as the network medium. For example, AppleTalk protocols can run on LocalTalk networks, Ethernet networks, and Token Ring networks. People typically describe a network in terms of the protocols it uses: an internet with Ethernet, Token Ring, and LocalTalk segments could form a single AppleTalk network.

Network addressing

Before you can call someone on the telephone, you need to know his or her telephone number. A person's telephone number represents the person's *address* in the telephone network — the place to which calls for that person should be directed.

Computers use network addresses for the same reason that callers use telephone numbers. When a network is set up, each *node* (device) on the network is assigned its own address. When one node wants to send a message to another, the network software requires the addresses of both the source node and the destination node.

On a simple network, a device's complete address can consist of a node number. On an internet, where a device belongs to one network among many, a device's address consists of the device's node number and the number of the network to which the node is connected.

The protocol used on a network determines the format for network addresses:

- ▶ TCP/IP uses a four-byte *dotted decimal* notation for internet (IP) addresses. IP addresses are described in more detail in "IP addressing" on page 3-3.
- ▶ AppleTalk uses a 16-bit number for network address information and an 8-bit number for node address information. AppleTalk addresses are described in more detail in "How AppleTalk works" on page 4-5.
- ▶ DECnet uses a sixteen-bit number to identify a node's DECnet area and address. DECnet addresses are described in more detail in "DECnet addresses" on page 5-1.

Binary, decimal, and hexadecimal numbers

To understand how network addresses work, you need to understand the relationship between binary, decimal, and hexadecimal numbers. For example, IP addresses are typically written as a set of four decimal numbers separated by dots (periods). To make sense of how one address is compared to another, however, you have to convert the dotted decimal number to its binary equivalent. For example, the IP address 192.32.222.10 can be written as the binary string 11000000 00100000 11011110 00001010.

Computers store information as a series of high and low electrical impulses. By representing a high impulse as a 1 and a low impulse as a 0, a computer converts the stream of electrical impulses to a series of 1's and 0's. Each 1 and 0 constitutes a *bit* (binary digit). Each set of eight bits constitutes one *byte*. Because computers work with bytes that are eight bits long, a single byte can range in value from 0 (00000000) to 255 (11111111).

Although computers process information as a string of 1's and 0's, users typically find binary notation cumbersome and difficult to remember. As a result, network addresses are typically converted from binary to decimal or hexadecimal notation:

- The **decimal numbering system** uses ten numerals (0 - 9) to represent numbers. Numbers greater than 9 are written as a string of digits, where each digit, reading from right to left, represent an incrementally greater power of 10. For example, the decimal number 234 means:

$$(2 \times 10^2) + (3 \times 10^1) + (4 \times 10^0)$$

or

$$(2 \times 100) + (3 \times 10) + (4 \times 1)$$

- The **binary numbering system** uses only two numerals (0 and 1) to represent numbers. Numbers greater than 1 are written as a string of digits, where each digit, reading from right to left, represent an incrementally greater power of 2. For example, the binary number 11101010 means:

$$(1 \times 2^7) + (1 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$$

or

$$(1 \times 128) + (1 \times 64) + (1 \times 32) + (0 \times 16) + (1 \times 8) + (0 \times 4) + (1 \times 2) + (0 \times 1)$$

or

$$234$$

- The **hexadecimal numbering system** uses 16 numerals (0-9 and A-F) to represent numbers. Numbers greater than 15 are written as a string of digits, where each digit, reading from right to left, represent an incrementally greater power of 16. For example, the hexadecimal number EA means:

$$(14 \times 16^1) + (10 \times 16^0)$$

or

$$(14 \times 16) + (10 \times 1)$$

or

234

Table 2-1 provides sample values in decimal, binary, and hexadecimal notation.

Decimal	Binary	Hexadecimal
0	0	0
1	1	1
2	10	2
4	100	4
10	1010	A
15	1111	F
16	10000	10
32	100000	20
255	11111111	FF

Table 2-1. Decimal/Binary/Hexadecimal conversion

Packets and datagrams

When a user sends a message over a network, the network software breaks the message into units called *packets*. As the network software prepares each packet for transmission, successive protocol layers add headers that contain formatting and addressing information to the data packet. A *datagram* consists of a packet and the header information needed to route the packet from one point to another over a network or internet.

When a datagram arrives at the destination computer, each protocol layer reads, processes, and removes its header information before passing the packet to the next layer. The network software then delivers the packet to the destination user in its original format.

Headers for packets on IP and AppleTalk networks contain source and destination address fields. The source address is always the address of the node sending the packet. The destination address is usually the address of a specific node that should receive the packet.

C

C

C

Chapter 3

TCP/IP and MacIP

What is TCP/IP?

IP addressing

IP subnetting

IP routing

What is MacIP?

NCSA Telnet

What is TCP/IP?

TCP/IP (Transmission Control Protocol/Internet Protocol) is a layered protocol suite (family) that lets computers from various manufacturers share information and resources across a network. Two computers that incorporate the TCP/IP protocol suite can exchange information across an internet, regardless of the connection method or vendor-specific issues.

TCP/IP was developed in the 1970's by the Department of Defense to promote a standard for connecting computer installations across the country. The TCP/IP standards have since been adopted by more than 250 vendors.

The TCP/IP protocol suite works like a library of routines that handle specific tasks relating to moving data from one host to another. Each protocol in the suite handles a different aspect of moving data from one host to another.

Internet Protocol (IP)

The Internet Protocol is responsible for sending packets from one network to another. IP accepts packets from TCP, UDP, or other transport protocols, and adds its own datagram routing information. IP then routes the packets through the appropriate routers or gateways to the message destination host. IP does not guarantee that packets will arrive in the proper sequence or that data will arrive intact at the destination machine.

Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) provides reliable data transport from one machine to another on a network. TCP accepts data from higher-level protocols, such as SMTP or TELNET. TCP segments the data into packets and adds a header containing control information, such as the packet source and destination ports and the packet checksum. TCP numbers each packet sequentially and passes it to the Internet Protocol. When IP delivers the packets to the destination machine, the destination machine's TCP sequences the packets, verifying that each one is intact and unduplicated. The TCP module on the destination host sends a numbered acknowledgment for each packet to the TCP module on the sender host. The TCP module on the sender host resends a packet if it does not receive a timely acknowledgment to its previous transmission, since it assumes that

the packet did not arrive intact at the destination host. If TCP cannot deliver data intact, in the correct order, on time, and without duplication, it informs the user that the connection to the remote TCP port cannot be maintained.

User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) provides data transport from a port on one host to a port on another host on a network. Like TCP, UDP accepts data from higher-level protocols, packages the data into packets, and passes the packets to the Internet Protocol for transmission across an internet.

Unlike TCP, however, UDP does not perform error checking or acknowledge when a packet is received. UDP is typically used when minimal protocol overhead is needed. Examples of UDP-based services include Trivial File Transfer Protocol (TFTP) services and network management query/response transactions.

File Transfer Protocol (FTP)

The File Transfer Protocol (FTP) lets a user on one host (the FTP client) interactively transfer files to and from another host (the FTP server) over a network or internet. FTP uses TCP to ensure that information arrives intact at the destination host.

TELNET

TELNET is an interactive remote access protocol that lets a user on one host log in (connect) to a remote computer on an internet. Once logged in, a user can control and monitor applications running on the remote host as if his or her terminal was connected directly to the remote host. TELNET uses TCP to ensure that information arrives intact at the destination host.



You should distinguish between the TELNET protocol and the NCSA Telnet application. NCSA Telnet, which is provided with the GatorBox, uses TELNET to let a Macintosh user connect to one or more IP hosts through the GatorBox. NCSA Telnet is described in more detail on page 3-20.

Internet Control Message Protocol (ICMP)

The Internet Control Message Protocol (ICMP) lets IP routers send error messages and network management information to other routers or hosts. For example, ICMP lets the IP software on one machine inform the IP software on another machine about an unreachable destination.

Simple Mail Transfer Protocol (SMTP)

The Simple Mail Transfer Protocol (SMTP) uses TCP to transfer and relay mail messages from one IP host to another over an internet. SMTP requires a front-end application to provide the user interface for mail transmission. GatorMail™ translates SMTP mail messages into a format usable to Macintosh mail applications.

IP addressing

Ethernet devices on a TCP/IP network use two types of addresses (hardware (or physical) addresses and IP (or logical) addresses) to identify themselves and share information. TCP/IP includes protocols (Address Resolution Protocol and Reverse Address Resolution Protocol) for mapping one address type to the other.

Hardware address

Every device on an Ethernet network has a built-in hardware address that is unique to that machine. Ethernet device vendors obtain a range of addresses from the SRI International Network Information Center (SRI-NIC). A vendor then assigns one number from its designated range to each device it manufactures. For example, the hardware address for every GatorBox starts with 00 00 89.

An Ethernet hardware address, which is typically written in hexadecimal format, is six bytes long: for example, 00 00 89 00 05 A2. The first three bytes identify the hardware manufacturer, while the last three bytes (00 05 A2) identify the specific device.

When a packet is sent over an Ethernet network, the packet header identifies the hardware addresses of the source and destination devices.

Internet (IP) address

Computers on a TCP/IP network communicate with one another using IP addresses. An IP address is a 4-byte (32-bit) number that identifies the network to which a computer is connected and the node number the computer is using on that network. A device with connections to more than one network requires a separate IP address for each connection.

Because a 32-character line of 1's and 0's is awkward to remember and use, IP addresses are typically broken into 8-bit segments, called bytes. Each byte is then converted to its decimal equivalent. When an IP address is written in decimal format, a period is inserted between bytes. For example, Cayman's internet address is 192.31.222.1, which is the decimal equivalent of the binary string 11000000 00011111 11011110 00000001. See "Binary, decimal, and hexadecimal numbers" on page 2-11 for more information on binary-decimal conversion.)

Network segment

Each IP address consists of a *network segment* and a *node segment*. The network segment of the IP address identifies the network to which the computer is connected. When one device on an internet wants to send a message to another device, it compares the network segment of its own IP address to the network segment of the destination IP address. If the two network numbers match, the device knows that it can contact the destination machine directly. If the network numbers do not match, the device knows it must direct the message to its network router or gateway to reach the destination machine.

In a sense, the network segment of an IP address functions like the area code for a telephone number. If you make a local call (that is, if the area code of the person you call matches your own area code), you can make the call without special procedures and your call does not require long-distance processing by the telephone company. If you make a long-distance call (that is, if the area code of the person you call does not match your own area code), you preface the telephone number with additional information (dialing a "1" and an area code), and the telephone company routes your call from your dialing area to the area of the person you are calling.

Node segment

The node segment of the IP address identifies a specific device on the designated network. Each device on a TCP/IP network must have a unique node number, which is typically issued by the network administrator.



You can verify that an IP address is not currently in use by issuing a ping [IP address] command on your IP network. If you receive an [IP address] is alive response, another device is using the IP address you specified.

Networks are frequently identified by replacing the node segment of the IP address with 0's. For example, a device with IP address 143.137.1.1 is said to be on the 143.137.0.0 network. This notation for identifying a network should not be confused with the network's broadcast address (see "Broadcasts and broadcast addresses" on page 3-6).

IP address classes

Unlike telephone numbers, where the area code has a fixed number of digits, IP addresses do not always use the same number of bits to identify a network. Instead, IP addresses are divided into classes:

- ▶ **Class A** — If a network will support a very high number of nodes, it uses IP addresses where only the first byte identifies the network and the remaining three bytes identify the node. The first node of a decimal Class A address is a number in the range 1-126. (Specifically, Class A addresses begin with 00 or 01 and use the last 24 bits of the 32-bit binary network address for the host number.) For example, IP address 1.2.3.4 identifies node 2.3.4 on network 1.
- ▶ **Class B** — If a network will support a moderate number of nodes, it uses IP addresses where the first two bytes identify the network and the remaining two bytes identify the node. The first node of a decimal Class B address is a number in the range 128-191. (Specifically, Class B addresses begin with 10 and use the last 16 bits of the 32-bit binary network address for the host number.) For example, IP address 140.2.3.4 identifies node 3.4 on network 140.2.
- ▶ **Class C** — If a network will support a small number of nodes, it uses IP addresses where the first three bytes identify the network and the remaining byte identifies the node. The first node of a decimal Class C address is a number in the range 192-223. (Specifically, Class C addresses begins with 11 and use the last 8 bits of the 32-bit binary network address for the host number.) For example, IP address 192.2.3.4 identifies node 4 on network 192.2.3.
- ▶ **Class D/Class E** — Specifications exist for Class D (IP multicast) and Class E IP addresses. However, they have not been widely implemented.



Sites interested in connecting to the Internet should obtain a range of IP addresses from the SRI Network Information Center (SRI-NIC).

Broadcasts and broadcast addresses

A broadcast is a message sent by one IP host to every other host on a network. Broadcasts are typically used when a host doesn't know how to reach a specific address or when one host wants to inform every other host about a problem.

When a device wants to send a broadcast, it uses a special *broadcast address* in place of a destination IP address or hardware address. The broadcast address for a network replaces the node bits of the IP address with 0's (old style) or 1's (new style). For example, the broadcast address for the unsubnetted network 192.31.222.0 would be 192.31.222.0 or 192.31.222.255.



BSD UNIX version 4.2, which many vendors incorporated into their software, used all 0's as the broadcast node address. When RFC 919 and RFC 922 were released in 1988, they standardized the use of 1's for a broadcast. BSD 4.3 adopted that standard. However, some vendors retain the old broadcast identifier.

A special form of broadcast address, 255.255.255.255, is used to send a broadcast to a device's local network. This broadcast address format is typically used when a sending host does not know its own IP address and wants to send a message to all hosts on its local network.

On many UNIX systems, you can identify the Ethernet interfaces for your host by using the `netstat -i` command (Figure 3-1). You can then identify the broadcast address for your network by using the `/usr/etc/ifconfig <interface>` command. For example, Figure 3-1 indicates that the Ethernet interface for host midway is `le0`, and that the broadcast address for `midway` is 192.31.222.0.

```

192.31.222.126
Use netstat -i to obtain the
Ethernet interface...

midway% netstat -i
Name Mtu Net/Dest Address Ipkts Ierrs ...then use ifconfig to obtain
le0 1500 192.31.222.0 midway 981443 367 the broadcast address.
lo0 1536 loopback localhost 172271 0
midway% /usr/etc/ifconfig le0
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
inet 192.31.222.126 netmask ffffffff broadcast 192.31.222.0
midway%

```

Figure 3-1. Using `netstat -i` and `ifconfig` to display a broadcast address

When a network implements subnetting, the decimal value for broadcast address bytes might not be 0 or 255. See “Broadcast addresses for subnet networks” on page 3-12.

Address Resolution Protocol (ARP)

When one device on a TCP/IP network wants to send a message to another device, it must know the destination device's hardware address. The Address Resolution Protocol (ARP) lets a TCP/IP host determine another host's 48-bit Ethernet hardware address if it knows the host's 32-bit IP address.

ARP assumes that every host knows the mapping between its own hardware address and its own IP address. The source host broadcasts an ARP request identifying its own hardware address and IP address and the IP address of the destination host. Every device on the source device's network receives the broadcast; if the destination device is on the same network, it responds directly to the source device with its hardware address.

Refer to RFC 826 for more information on the IP address resolution protocol.



ARP cache

To minimize the number of broadcasts traveling over the IP network, IP hosts, routers, and gateways maintain a cache of IP-to-physical-address mappings. When an IP host receives a new IP address/hardware address pair, it saves the information in this ARP cache. When the host wants to send a packet, it checks its ARP cache for an address pair before it sends an ARP request. To prevent filling the ARP cache with obsolete entries, each ARP cache is associated with a timer, and entries not used within a fixed period are deleted.



On many UNIX systems, you can view the ARP cache for a host by using an `/usr/etc/arp -a` command.

Proxy ARP

Proxy ARP is a variation on the ARP protocol that lets a router respond to ARP requests on behalf of devices on another (concealed) network. Instead of forwarding an ARP request to devices on its network and letting a device respond for itself, the router determines whether the correct device is on its network; if it is, the router sends back its own hardware address in an ARP

reply, saying, in effect, "I am that device." The sending host maps the remote host's IP address to the router's hardware address, and subsequently forwards all communication for the remote host to the router.

Proxy ARP lets a network administrator add network segments to an internet without modifying routing tables on other hosts or routers on the network. Proxy ARP in GatorKeeper is typically used when the subnet mask for the GatorBox's LocalTalk IP network is more restrictive than the subnet mask for the rest of the IP internet.

Reverse Address Resolution Protocol (RARP)

The Reverse Address Resolution Protocol (RARP) lets a TCP/IP host determine its own IP address (or another host's IP address) if it knows the its 48-bit hardware address. RARP is most frequently used by diskless workstations, which know their hardware address but not their IP address when they are turned on.

RARP requires that one or more server hosts maintain a database that maps hardware addresses to IP addresses, and that this RARP server respond to requests from RARP clients. The RARP client broadcasts a RARP request identifying its hardware address. The RARP server uses the hardware address to identify the client's IP address. The RARP server then sends a RARP reply with the IP address that corresponds to the hardware address to the client.



Refer to RFC 903 for more information on the Reverse Address Resolution Protocol.

IP subnetting

IP subnetting lets an organization partition one physical IP network into two or more logically separate networks. Subnetting uses some of the node bits in the 32-bit network address as additional network information. Because Class C networks provide only 254 node addresses, sites with several hundred nodes can obtain a Class B network number (which provides 65535 node addresses) and partition the Class B network into discrete Class C subnets.

For example, assume that San Dimas University is setting up a class B network (143.137.0.0) to connect several hundred nodes spread over four

buildings. The network administrator could issue sequential IP numbers to devices as needed: 143.137.0.1, 143.137.0.2, etc. However, since devices with similar IP addresses would be in different buildings and devices with very different IP addresses would be next to each other, sequential numbering would complicate network administration.

A better network numbering scheme would assign nodes in the same building addresses with the same number in the third byte. For example, all nodes in the Administration building might be assigned IP addresses of 143.137.10.### (where ### represents a unique node number for that building), while nodes in the Chemistry building would have IP addresses of 143.137.20.###.

While this numbering scheme organizes IP addresses into logical groups, it is still inefficient from the network's perspective. Since all devices at San Dimas University still belong to the same IP network, every broadcast from every host must be processed by every other host. A significant portion of each host's processing time would therefore be spent on network overhead.

Subnetting the San Dimas network into separate IP networks for each building solves this problem. The network administrator would set up a subnet mask to specify that the third byte of a device's IP address represents additional network number information instead of node number information. This subnet mask would let the San Dimas University network support as many as 255 subnets, each of which can support as many as 254 nodes.

Figure 3-2 illustrates the network map for the San Dimas University network. The Class B network is subnetted into five Class C subnetworks: 143.137.10.0 (Administration Building), 143.137.20.0 (Chemistry Building), 143.137.30.0 (Mathematics Building), 143.137.40.0 (English Building), and 143.137.100.0 (Ethernet backbone). Each building functions as an independent network. If a host in one building issues a broadcast message to its local network, hosts in the other buildings do not receive it.

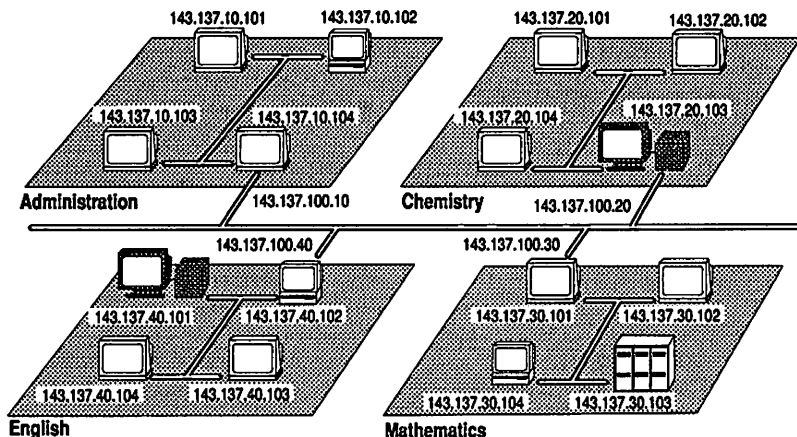


Figure 3-2. Subnetting the San Dimas University network

The hosts that route to the backbone have two IP addresses: one for the connection to the backbone subnet and one for the connection to the building subnet. When the router receives a packet intended for a host on another subnet, it will forward the packet to the appropriate router. For example, if 143.137.10.101 requests a TELNET connection to 143.137.30.102, the router in the Administration Building would forward the TELNET request to the router in the Mathematics Building, which would forward the request to the appropriate host.

Subnet mask

The *subnet mask* specifies which bits of the 32-bit IP address represent network information. The subnet mask, like an IP address, is a 32-bit binary number: a 1 is entered in each position that will be used as network number information and a 0 is entered in each position that will be used as node number information. When a host on the subnet compares its network number to that of a destination host (to determine whether it must forward the message to a router or whether it can contact the host directly), it will use each bit specified by the subnet mask to make the comparison.

It may not be obvious that IP hosts always perform subnet matching, even when a subnet mask is not specified. When a subnet mask is not explicitly

established for a network, the IP host uses the default subnet mask for the network class:

- ▶ **Class A:** 255.0.0.0 (11111111 00000000 00000000 00000000)
- ▶ **Class B:** 255.255.0.0 (11111111 11111111 00000000 00000000)
- ▶ **Class C:** 255.255.255.0 (11111111 11111111 11111111 00000000)

To see how a subnet mask would be used, assume that host 143.137.10.101 in the Administration Building wanted to send a message to host 143.137.30.102. Before it can send a message to the destination host, the sending host must determine whether it is on the same network. Figure 3-3 illustrates how a host in the Administration Building would perform the comparison if the standard subnet mask for a Class B network (255.255.0.0) is used. As Figure 3-3 indicates, the sending host would use the subnet mask to determine that only the first 16 bits (two bytes) of the IP addresses should be used for the network address comparison. Since the first two bytes of the IP addresses match, the sending host knows that it and the destination host are on the same network and that, as a result, it can contact the destination host directly.

Subnet mask:	11111111	11111111	00000000	00000000
Sender IP address (143.137.10.101):	10001111	10001001	00001010	01100101
Destination IP address (143.137.30.102):	10001111	10001001	00011110	01100110
	Match	Match		

Figure 3-3. Comparison of network numbers with default Class B subnet mask

Figure 3-4 illustrates how the same host would perform the comparison after a more restrictive subnet mask is implemented. The subnet mask indicates that the first 24 bits (three bytes) of the IP addresses should be used for the network address comparison. Although the first two bytes of the IP addresses match, the third byte of the two addresses does not. Consequently, the sending host knows that it and the destination host are on different networks and that it must work through an IP router to contact the destination host.

Subnet mask:	11111111	11111111	11111111	00000000
Sender IP address (143.137.10.101):	10001111	10001001	00001010	01100101
Destination IP address (143.137.30.102):	10001111	10001001	00011110	01100110
	Match	Match	No match	

Figure 3-4. Comparison of network numbers with Class C subnet mask

The preceding example discussed subnet masks in terms of contiguous 8-bit segments (bytes). A subnet mask can actually use part of a byte (or even discontinuous bits) to identify additional network information. Using a complete byte to identify a subnet is a matter of convenience more than technical necessity. For example, assume that a network has hosts on three networks (143.137.222.101, 143.137.223.31, and 143.137.224.86) and a subnet mask of 11111111 11111111 11100000 00000000. Determining which hosts are on the same subnet is far from obvious.

To determine which hosts are on the same subnet, a router would convert the decimal IP addresses to their binary equivalents and compare the first 19 bits. Figure 3-5 illustrates that hosts 143.137.222.101 and 143.137.223.31 are on the same subnet, but that 143.137.224.86 is on a different subnet. In essence, given the 255.255.224.0 subnet mask, node 143.137.222.101 has a node number of 30.101 on network 143.137.192.0.

Subnet mask:	11111111	11111111	11100000	00000000
143.137.222.101:	10001111	10001001	11011110	01100101
143.137.223.31:	10001111	10001001	11011111	00011111
143.137.224.86:	10001111	10001001	11100000	01010110

Figure 3-5. Subnet mask using partial byte

Broadcast addresses for subnet networks

Setting up a subnet mask for a network changes the broadcast address for that network. If the subnet mask uses a complete byte as additional network information, you must include that byte in the network segment of the broadcast address. For example, the broadcast address for network

143.137.0.0 (without a subnet mask) would be 143.137.255.255. If you use the third byte of the address for network information, the broadcast address for the 143.137.10.0 subnet would be 143.137.10.255.

If the subnet uses part of an address byte for the network address, you must include the network address bits in the broadcast address for the subnet. This will result in broadcast address nodes other than 255. For example, if you set up a subnet mask of 255.255.224.0

(11111111 11111111 11100000 00000000) for network 143.137.0.0, the broadcast address for subnet 143.137.10.0 would be 143.137.31.255, while the broadcast address for subnet 143.137.222.0 would be 143.137.223.255.

Subnetting a LocalTalk network

When a LocalTalk network behind a GatorBox is configured as an extension of an IP network, the GatorBox can reserve as many as 64 IP addresses for devices on the LocalTalk network that require IP addresses, such as a Macintosh running NCSA Telnet. By configuring the LocalTalk network as a separate IP subnet, you reserve the entire range of node addresses for MacIP assignment. For more information about MacIP address assignment, refer to "Dynamic MacIP address assignment" on page 3-16.

IP routing

When an IP host transmits information to another device on its own network, it identifies the device's hardware address and begins sending it packets. If the IP host wants to transmit information to a host on a different network, it sends packets to an IP router. If the router has a direct connection to the destination host's network, it transmits packets directly to the destination host. If the router is not connected to the destination host's network, the packet is forwarded from one router to another until it reaches a router connected to the destination host's network.

Routing tables

IP routers maintain routing tables that identify how to reach remote networks. Each entry in the routing table lists a network known to the router, the distance (in hop counts) to the remote network, and the router to which packets for the remote network should be forwarded.

For example, Figure 3-6 illustrates the routing table for Router 1 on an IP internet. The routing table indicates that Router 1 is connected directly (Distance = 0) to the 192.31.222.0 and 192.31.223.0 networks and that Router 2 (192.31.223.21) is the router to the 192.31.224.0 network.

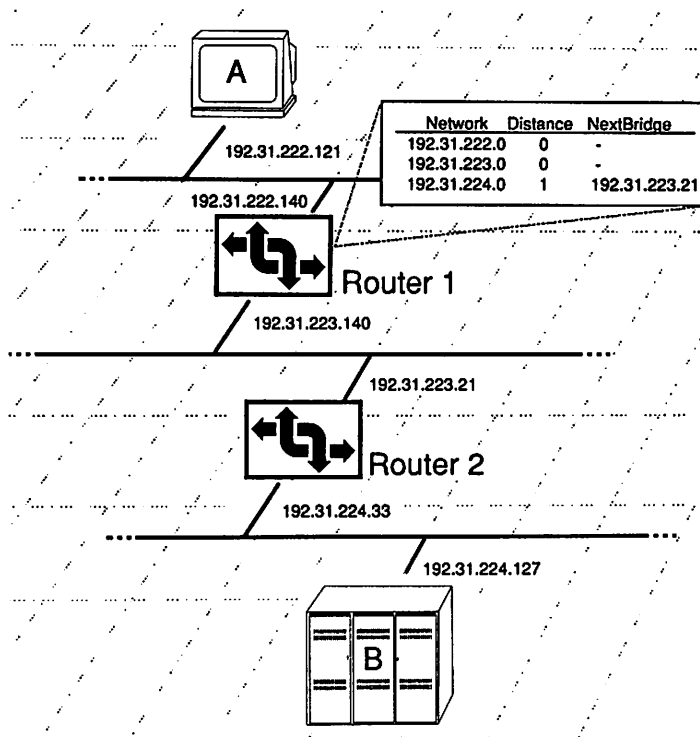


Figure 3-6. Routing table

When a router receives a packet, it compares the network segment of the destination IP address to each entry in its routing table to determine whether it can reach the destination host directly or whether it must forward the packet to another router. If device A in Figure 3-6 (192.31.222.121) wanted to send a packet to device B (192.31.224.127), it would send the packet to Router 1. Router 1 would compare the packet's destination address to the entries in its routing table. Since the distance to device B's network is greater than 0, Router 1 knows device B is on a remote network. It would therefore forward the packet to Router 2, which would look in *its* routing table, verify that the destination host is on one of its networks, and send the packet to device B.



On many UNIX systems, you can use the `netstat -r` command to display the routing table for an IP host.

Routing Information Protocol (RIP)

RIP (Routing Information Protocol) lets IP routers automatically share information about routes to other networks.

- ▶ If a gateway *actively* supports RIP, it broadcasts information about its routing table every 30 seconds. Each RIP broadcast message identifies the IP networks that the router can reach and the distance to the router for that network. Other routers can use the RIP information to update their own routing tables.
- ▶ If a gateway *passively* supports RIP, it doesn't broadcast information about its own routing table, but it updates its routing table with information supplied by active gateways.

If your LocalTalk network is set up as an IP subnet, you can specify whether the GatorBox will broadcast RIP packets to other gateways (that is, act as an active router) and whether the GatorBox will accept RIP messages from other gateways (that is, act as a passive router).

- ▶ If you specify that the GatorBox **can accept but not broadcast** RIP packets about its LocalTalk subnet, you must specify a route to the subnetted LocalTalk in the routing table of all IP routers and hosts on the Ethernet wire. Without the manual specification of the route, IP hosts on the Ethernet network have no way of knowing how to get to the subnetted LocalTalk network.
- ▶ If you specify that the GatorBox **cannot** accept RIP packets, the GatorBox cannot automatically update its routing table. Consequently, all packets traveling through the GatorBox from the LocalTalk network to a remote Ethernet network would be directed to the default gateway specified for the GatorBox, leaving it to the default router to direct the packets to the correct destination.

If the GatorBox receives a redirect message from the default gateway while trying to reach a remote host, the GatorBox will direct packets to the address provided by the default router instead of to the default router itself.



On many UNIX systems, you can use the `route add` command to add routes to the routing table on most UNIX systems. The syntax for the `route add` command is:

```
route add <destination network> <IP router> <hop count>
```

For example, you would use the command `route add 140.100.10.0 140.100.5.20 1` to tell an IP host that packets destined for network 140.100.10.0 should be sent to the GatorBox at address 140.100.5.20.

For information on how to configure the GatorBox to support RIP, refer to the *GatorBox User's Guide*.

What is MacIP?

MacIP is an AppleTalk protocol that lets a Macintosh communicate with a TCP/IP network through a GatorBox. MacIP lets Macintoshes encapsulate IP packets inside AppleTalk DDP (Datagram Delivery Protocol) packets. When the Macintosh sends these IP-inside-DDP packets to the GatorBox, the GatorBox strips off the DDP encapsulation and forwards the IP packet to the appropriate IP host on the internet. When the IP host sends a reply, the GatorBox encapsulates the IP packet in DDP and forwards it to the Macintosh.

A Macintosh running an application that uses the IP protocols, such as NCSA Telnet or MacTCP, must be assigned an IP address to communicate with IP hosts. The GatorBox reserves a range of addresses for assignment to LocalTalk devices (the MacIP address range). When the GatorBox receives a packet addressed to an IP address in its MacIP range, it responds (proxies) for the address.

Dynamic MacIP address assignment

Dynamic address assignment lets the GatorBox assign an IP address to a LocalTalk node from a range of IP addresses reserved for that purpose. Because a GatorBox can, over time, assign the same IP address to several Macintoshes, dynamic address assignment requires that fewer IP addresses be reserved for MacIP and provides greater flexibility in IP address assignment. However, dynamic address assignment can complicate network management, since an IP address is not assigned to a specific device.

When the GatorBox is configured to support MacIP, it registers itself on its LocalTalk network with an NBP device type of IPGATEWAY. When a

Macintosh runs an application that requires an IP address, such as NCSA Telnet or MacTCP, the Macintosh locates its IP gateway by issuing an NBP Lookup for `=:IPGATEWAY@*` (which means "Is there an IP gateway in my zone?"). When the GatorBox responds with an NBP Reply in the format `<GatorBoxName>:IPGATEWAY@*` (meaning "I am the IP gateway for your zone"), the Macintosh issues a `GetIPAddress` command to the GatorBox. The GatorBox issues the first nonactive number in its dynamic address range to the Macintosh. When it receives its IP address, the Macintosh issues an NBP Lookup command (`<IP address>:IPADDRESS@*`, meaning "Is any device in my zone using `<IP address>`?") to verify that the address it has been assigned is not already in use. A response to the NBP Lookup would indicate that another node is using the same IP address. If no device responds to the NBP Lookup, the Macintosh assumes that the assigned IP address is safe to use. (Refer to "Name Binding Protocol (NBP)" on page 4-2 for an explanation of NBP Lookup and NBP Reply transactions.)

The GatorBox maintains a table mapping each IP address it assigns to a device on its LocalTalk network to an AppleTalk network address. Once an address has been assigned, the GatorBox periodically queries the LocalTalk node to confirm that it is still using the address.

Static MacIP address assignment

Static address assignment means that a Macintosh uses the same IP address each time it runs an application requiring an IP address. A user typically obtains a static MacIP address from the network administrator. The user then specifies that address when he or she configures the application using MacIP. Strictly speaking, a GatorBox doesn't *assign* a static MacIP address. Rather, the GatorBox *protects* the addresses in its reserved static address range to prevent other devices from using them.

Static MacIP address assignment will not work through an AppleTalk router, such as a Hayes InterBridge or a Shiva NetBridge, if the LocalTalk network behind the router has a different zone name. Because a Macintosh configured to use a static MacIP address does not actively solicit the IP address from the GatorBox, the GatorBox does not have the opportunity to map the Macintosh's IP address to its AppleTalk address initially. The GatorBox does locate nodes on its LocalTalk network that are using static MacIP addresses by issuing an NBP Lookup (`=:IPADDRESS@*`, which means "What devices in my zone are using IP addresses?") and then mapping the static IP addresses of nodes that reply to their AppleTalk addresses. If the LocalTalk network on the other side of the AppleTalk router has a different

zone name, a device on the network does not receive or reply to the GatorBox's NBP Lookup. Consequently, the GatorBox cannot map the IP address being used by the remote node to its AppleTalk address.

MacIP on subnetted LocalTalk

Setting up the LocalTalk network behind a GatorBox as a separate IP subnet changes the manner in which MacIP distributes IP addresses. When you specify that the LocalTalk network is a separate subnet, the GatorBox reserves all the numbers in that subnet for itself. The network administrator then specifies how many IP addresses on that subnet are reserved for dynamic MacIP requisition. Any address on the LocalTalk subnet outside the range reserved for dynamic assignment is available for static MacIP address assignment.

AppleTalk Address Resolution Protocol (AARP)

The AppleTalk Address Resolution Protocol (AARP) maps a foreign address, such as an Ethernet hardware address, to an AppleTalk network/node number. AARP maintains an Address Mapping Table (AMT) that matches the foreign address type to its corresponding AppleTalk address. When an AARP client requests an address for a destination node, AARP checks the AMT to see if the address map has an entry for the destination. If it does, AARP returns the address to the client. If it does not, AARP broadcasts a request packet asking for the hardware address corresponding to the designated AppleTalk address.

The GatorBox supports two types of AppleTalk address resolution:

- **NBP-style address resolution** uses the Name Binding Protocol to identify the AppleTalk address associated with an IP address. When NBP-style AARP is used, the GatorBox broadcasts an NBP Lookup in the format `<IP_Address>:IPADDRESS@<zonenumber>` to its LocalTalk network. For example, if the GatorBox is trying to locate the device with IP address 192.31.222.110, it would issue an NBP Lookup of `192.31.222.110:IPADDRESS@*`. NBP-style AARP lets the GatorBox offer MacIP support to all of the LocalTalk and EtherTalk networks within a given AppleTalk zone.

- **DDP-style address resolution** uses the Datagram Delivery Protocol to identify the AppleTalk address associated with an IP address. Because DDP address resolution does not traverse the zones, use of DDP-style AARP limits address lookups to the LocalTalk network directly connected to the GatorBox. DDP-style AARP does not allow the GatorBox assign IP addresses dynamically in NCSA Telnet. DDP-style ARP has been largely replaced by NBP-style ARP. Support for DDP-style ARP is included for sites that still use older versions of applications such as Brown University's tn3270.

Restricting MacIP services to LocalTalk

When a Macintosh on LocalTalk begins using an IP address, it registers its address on the network to verify that no other device is using the same IP address. When two or more AppleTalk networks on an internet share the same zone name, an attempt to register an IP address can result in a misleading response from other routers, which will incorrectly claim the address as their own. Restricting MacIP services to LocalTalk prevents a GatorBox from responding to a registration request from Macintoshes on other AppleTalk networks.

Figure 3-7 illustrates a situation where restricting MacIP services to LocalTalk is necessary. Assume that a site has two GatorBoxes (GatorBox1 and GatorBox2) on the same EtherTalk network. GatorBox1 and GatorBox2 are each set up with a range of MacIP addresses: GatorBox1 has a range of 100.0.0.5 to 100.0.0.10 and GatorBox2 has a range of 100.0.0.15 to 100.0.0.20. Although the numbers for the LocalTalk networks connected to the two GatorBoxes are different, GatorBox1 and GatorBox2 share the same AppleTalk zone name (Admin Zone).

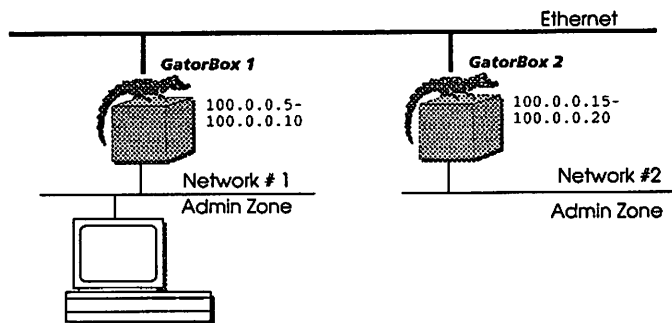


Figure 3-7. Restricting MacIP services to LocalTalk by limiting NBP Lookups

When the Macintosh behind GatorBox 1 runs NCSA Telnet, it obtains a dynamic IP address (100.0.0.5) from the GatorBox reserved address range. The Macintosh then issues an NBP Lookup broadcast ("Is any device in my zone using <IP address>?") to its zone to verify that the IP address is available.

Because GatorBox 1 and GatorBox 2 are in the same AppleTalk zone (Admin Zone), GatorBox 1 passes the NBP Lookup to GatorBox2 for forwarding to Network 2. However, instead of forwarding the NBP Lookup, GatorBox 2 will send an NBP Reply to the Macintosh claiming the 100.0.0.5 address and forcing the Macintosh to look for another IP address. Why? Any MacIP gateway, including the GatorBox, will always answer registration requests in its AppleTalk zone for IP numbers **outside** its MacIP range. When the IP address is outside its reserved IP range, the GatorBox assumes that a device in its zone is trying to telnet to that address and says, in effect, "I am your gateway to the machine with that address. Therefore direct your transactions to me." Since the reserved MacIP address ranges in the two GatorBoxes are mutually exclusive (as they should be), each GatorBox would respond to registration attempts from Macintoshes on the other GatorBox's LocalTalk network, preventing anyone from running Telnet.

Restricting NBP Lookups to the LocalTalk network on which they originate resolves this problem. As long as another device on its LocalTalk network does not respond to its NBP Lookup, the Macintosh can obtain an IP address from its GatorBox and safely register it on its own network.

NCSA Telnet

The NCSA Telnet application uses the TCP/IP TELNET protocol to provide interactive access from a Macintosh to IP hosts. With NCSA Telnet, a Macintosh user can maintain simultaneous connections to several TCP/IP hosts. NCSA Telnet includes a standard file transfer protocol (FTP) that lets users transfer files to and from other remote machines.



On many UNIX systems, you can verify that the TELNET daemon is running by using a ps -aux command and looking for telnetd.



The following discussion is drawn from the NCSA Telnet documentation, which is on the Network Applications disk supplied with your GatorBox software. For more information about NCSA Telnet, refer to the NCSA Telnet manuals.

Standard Telnet and MacTCP Telnet

Starting with version 2.4, NCSA Telnet comes in two versions—a standard version and a MacTCP version:

- ▶ The standard version, called “NCSA Telnet 2.4”, uses drivers built into the application to communicate with the TCP/IP network. The standard version of NCSA Telnet may conflict with other applications that use MacTCP to access the TCP ports.
- ▶ The MacTCP version, called “NCSA Telnet 2.4—MacTCP—”, uses the Apple MacTCP drivers and Hosts file to communicate with the TCP/IP network.

The Network Applications disk that comes with the GatorBox software contains both versions of the NCSA Telnet application, together with their configuration files and documentation.

NCSA Telnet Settings File

NCSA Telnet lets you establish default settings to control terminal emulation settings. These settings are stored in the NCSA Telnet Settings file, which is stored in the System Folder.

NCSA Telnet config.tel file

The NCSA Telnet configuration file (config.tel) is a text file that contains standard operating parameters and a list of commonly accessed hosts. NCSA Telnet reads the config.tel text file when it starts to obtain its default configuration settings. Because NCSA Telnet only reads the config.tel when the program starts, you must restart Telnet to make changes to the config.tel effective.

The configuration settings you must update in the config.tel file depend on the version of NCSA Telnet (MacTCP or non-MacTCP) you run. For example, you specify name servers and gateways in the config.tel file for the non-MacTCP version, but the MacTCP version relies on the MacTCP Hosts

file (in the System Folder) to obtain information about zones, nameservers, and gateways. Fields in the sample config.tel file that the MacTCP version ignores are noted.

config.tel syntax

The config.tel file consists of a series of *keywords* and *values*. You should observe the following syntax guidelines when you set up config.tel file entries:

- ▶ Each keyword/value pair can be placed on a separate line. Alternatively, you can put related keyword/value pairs on the same line.
- ▶ Keywords and values must be separated by a colon (:), semicolon (;), equal sign (=), or whitespace character. To include delimiters in a value field, enclose the field in double quotes.
- ▶ Text strings must be enclosed in double quotes. Quotes cannot be a part of any value field.
- ▶ Comments begin with a pound sign (#). A comment can begin at the start of a line or in the middle of a line.

Sample config.tel file

Figure 3-8 illustrates the config.tel file supplied with the GatorBox software. A brief discussion of each config.tel file entry follows.

```

# Example host file for NCSA Telnet 2.4
#
# Note: When running the MacTCP drivers, the following fields are meaningless,
# but may be included for compatibility with systems which do not have MacTCP:
#     hardware,gateway,arptime,rwin,mtu,maxseg,retrans,zone
#
# "funny, this configuration file is readable..."
#
# This file is free form
# Separators are any char <33 and ;:=
#
# The form is keyword=value for each parameter.
# The first set of parameters refer to the whole program's defaults.
# These parameter values can be in any order.
# Following this are the individual machine specs.
# If the first machine is name "default", then it contains default
# values for the rest of the machines.
#
hardware=AppleTalk                # Network connection type:
                                   #   values are: AppleTalk, Ether
                                   #   Ether<n>, EtherSE, EtherSC
#zone="Zone"                      # Which zone is the gateway in? (AT only)
ftp=yes                          # do you want ftp enabled?
#domain="ncsa.uiuc.edu"          # default domain for name lookups
termtype="vt100"                # terminal type to advertise (safe bet)
arptime=5                       # arp timeout in seconds
                                   #   affects machines on your local network
block = 200                     # screen blocking factor, range: [100 to 4000]
#passfile="ftppass"             # name of file for FTP passwords in (System Folder)
commandkeys=yes                 # Yes, I want command keys as default
#
# Following are individual machine specifications
#
# The machine named "default" contains the fields which are automatically
# filled in for other hosts. name=default machine should appear first.
#
name = default                  # Session name, "default" is a reserved name
                                   #   Not a real machine, default parameters only
#host=sri-nic.arpa              # Actual host name of machine, not session name
#hostip=10.0.0.51               # IP address of host, example is for SRI-NIC
#gateway=1                      # This machine is a gateway for me
#nameserver=1                   # This machine has a DOMAIN name server for me
scrollback=200                  # number of lines of scrollback per session
erase=delete                    # use delete code or backspace code for <- key?
                                   #   legal values are "delete" and "backspace"
vtwrap=yes                     # should VT100 be in wrap mode or not?
vtwidth=80                     # 80 or 132 columns on startup of session

```

Figure 3-8. Sample config.tel file for NCSA Telnet 2.4

```

#vtlines=24                # number of lines, 24 is standard VT102
#port=23                   # choose the TCP port for telnet (std = 23)
nfcOLOR="{0,0,0}"          # normal, foreground (Mac II)
nbcolor="{65535,65535,65535}" # normal, background (Mac II)
bfcolor="{0,0,0}"          # blink, foreground (Mac II)
bbcolor="{65535,65535,65535}" # blink, background (Mac II)
#font="Courier"            # font and size, default is Monaco 9
#fsize=12
#crmap=4.3BSDCRNUL         # map of the CR key for compatibility
#duplex=half               # modifier for non-echo mode, forces send
clearsave=yes              # save lines on clear screen yes/no
# The following entries affect the tuning of TCP connections to this host.
# They should be set by the network administrator who is familiar with
# the requirements of your specific network.
contime=20                 # timeout in seconds to try connection
retrans=30                 # starting retransmit time out in ticks
# 1/60ths of sec
mtu=512                    # maximum transmit unit in bytes
# AppleTalk MAX = 512, Ethernet Max=1024
maxseg=512                 # largest segment we can receive
# AppleTalk MAX = 512, Ethernet Max=1536
rwin=512                   # TCP window size, MAX=4096
#
# Below this line, most of the communication parameters are obtained
# from the "default" host entry.
# Machine names, IP addresses, and special communication parameters are
# present when needed.
#

#name=mynameserver ; hostip=127.0.0.2 ; nameserver=1
#name=mygateway ; hostip=127.0.0.3 ; gateway=1

name=ncsaa ; hostip=128.174.3.100
name=ncsab ; hostip=128.174.3.101
name=ncsad ; hostip=128.174.10.48

```

Figure 3-8 (continued). Sample config.tel file for NCSA Telnet 2.4

Entries in the sample config.tel file include:

- ▶ **hardware=** specifies the type of Ethernet device the Macintosh uses to connect to the TCP/IP network. **hardware=AppleTalk** specifies that the Macintosh will work through a DDP-to-IP gateway (the GatorBox). This field is not used for the MacTCP version of NCSA Telnet.
- ▶ **zone=** specifies the zone in which the gateway to the TCP/IP network resides. In some situations, an AppleTalk to Ethernet gateway may be used even if that gateway is not in the local AppleTalk zone. This field is not used for the MacTCP version of NCSA Telnet.

- ▶ `ftp=` specifies the default value for FTP (file transfer) functionality. Values are yes and no. The `ftp=` setting can be overridden from within an NCSA Telnet session.
- ▶ `domain=` specifies the default domain for name lookups. If a domain request does not contain a period (.), the domain suffix is appended to the request before it is sent to the nameserver.
- ▶ `termtype=` specifies the type of terminal (or terminal emulation) to be used to communicate with the TCP/IP network. `termtype="vt100"` specifies that the Macintosh will emulate a DEC VT-100 terminal.
- ▶ `arptime=` specifies the time, in seconds, that the Macintosh should continue trying to reach a host on the local wire.
- ▶ `block=` specifies the number of text characters, in the range 100-4000, that NCSA Telnet will read from the network as a block. A low block value means faster turnaround on typed commands, such as Control-C. A high block value means better overall throughput to the screen.
- ▶ `timeslice=` (not included in the default `config.tel` file) specifies the amount of time you are willing to wait to process information. This option is useful when you are using MultiFinder, as it lets you run other programs in the background. The default is 3, meaning "three Macintosh clock ticks." **You should set the `timeslice=` setting to 2 if you notice Telnet connections being dropped after short periods of inactivity.**
- ▶ `passfile=` specifies the name of the Macintosh file in which FTP usernames and passwords are listed. If a `passfile` is specified, users are prompted to enter their name and password before they can transfer files to or from the Macintosh. If a `passfile` is not specified, Telnet does not perform password checking.
- ▶ `commandkeys=` specifies whether NCSA Telnet should support the Macintosh command-key equivalents for menu entries (such as Command-C for Copy). Values are yes and no.
- ▶ `name=` specifies the name assigned to the configuration settings for a Telnet session. You can create one or more session configurations for each host to which you connect by creating multiple `name=` entries and following each entry with the appropriate session settings.

The first name = entry is typically name=default. Configuration settings that follow name=default apply to other sessions unless they are specifically overridden.

- ▶ host= specifies the actual name of the host for which the session is configured. If NCSA Telnet cannot find a session with the name specified in an Open Connection request in the config.tel file, it attempts to open a connection to a host with the specified name.
- ▶ hostip= specifies the IP address of the host. IP addresses of gateways and nameservers must be in the config.tel file.
- ▶ gateway= specifies whether the host is a gateway to other networks. If more than one gateway is available, specify the sequence in which NCSA Telnet should check for gateways by incrementing the gateway= number.

You must enter a hostip= keyword if a host is a gateway. For example, you could enter name=GatorBox103456; hostip=143.137.10.1; gateway=1 to specify that the designated GatorBox at IP address 143.137.10.1 is the gateway between your LocalTalk and Ethernet networks. Not used by the MacTCP version of NCSA Telnet.



If you use Telnet from a LocalTalk network set up as a separate IP subnet, you must specify the GatorBox as the gateway in your config.tel file. For example, you would include the config.tel entry:

```
name=GatorBox00NNN; hostip=140.100.10.20; gateway=1
```

to specify that the GatorBox called GatorBox00NNN at LocalTalk IP address 140.100.10.20 is the primary gateway for Telnet connections.

- ▶ nameserver= specifies that a host is a domain nameserver. If NCSA Telnet cannot find a session or host in the config.tel file with the name specified in an Open Connection request, it uses UDP to query a domain nameserver for host names.

If more than one nameserver is available, specify the sequence in which NCSA Telnet checks nameservers by incrementing the nameserver= number. When the domain request to one nameserver times out, NCSA Telnet sends the nameserver query to the next nameserver until the maximum number of retries is reached or a response is received. You must enter a hostip= keyword if a host is a nameserver.

- ▶ **scrollback=** specifies the number of lines NCSA Telnet will retain for this session. Each scrollback line requires at least 86 bytes of memory.
- ▶ **erase=** specifies whether the backspace key will send a delete or backspace command to the host. Values are **erase=delete** and **erase=backspace**.
- ▶ **vtwrap=** specifies whether characters that run beyond the right screen margin should be wrapped to the next line. Values are **yes** and **no**. If you specify **vtwrap=no**, each new character printed to the screen replaces the last character on the current line and the cursor does not move.
- ▶ **vtwidth=** specifies the number of characters that can be displayed on a line. Values are 80 and 132.
- ▶ **vtlines=** specifies the number of lines that can be displayed in the window for the terminal session (and, by extension, the default height for the session window). The standard value for a VT102 window is 24 lines.
- ▶ **port=** specifies the TCP port number to use when connecting for this session. The Internet standard port number for the TELNET protocol is 23.
- ▶ **nfcOLOR=** specifies the normal foreground color for the terminal session. The format of the color specifier is {red, green, blue}, where red, green, and blue are the integer numbers corresponding to the requested colors. Color specifications are ignored if the terminal session is run on a monochrome Macintosh.
- ▶ **nbcolor=** specifies the normal background color for the terminal session.
- ▶ **bfcOLOR=** specifies the blinking foreground color for the terminal session.
- ▶ **bbcolor=** specifies the blinking background color for the terminal session.
- ▶ **font=** specifies the default font for the terminal session. The font name must exactly match the name of a font in the Macintosh System file.

- ▶ **fsize=** specifies the default font size (in points) for the terminal session.
- ▶ **crmap=** specifies the carriage return (end-of-line) character for the terminal session. Values are `4.3bsdcrnul` (carriage return only) and `CRLF` (carriage return/line feed).
- ▶ **duplex=** specifies the echo mode setting. This parameter only applies to hosts that negotiate non-echoing mode but do not expect local line editing. Values are `half` (all character keys are sent and echoed to the screen immediately) and `full` (characters are echoed locally and queued until a RETURN or CONTROL character is sent). The **duplex=** setting has no effect when local echo is off.
- ▶ **clearsave=** specifies whether the screen is saved to the scrollbar region when a clear screen command is received. Values are `yes` and `no`.
- ▶ **contime=** specifies the time, in seconds, NCSA should attempt to open a connection to the specified host. Does not apply to the MacTCP version of NCSA Telnet.
- ▶ **retrans=** specifies the initial retransmission timeout, in 60ths of a second, for a terminal session. Increasing the value of this parameter may help in reducing the initial burst of retries that is typical of connections with high round-trip times.
- ▶ **mtu=** specifies the maximum number of bytes that can be included in a transmitted packet. The maximum transmission unit for a session originating from LocalTalk is 512. The MTU for an EtherTalk session is 1024. Does not apply to the MacTCP version of NCSA Telnet.
- ▶ **maxseg=** specifies the maximum number of bytes that can be included in a received packet. The maximum segment size for a session originating from LocalTalk is 512. The MTU for an EtherTalk session is 1536. Does not apply to the MacTCP version of NCSA Telnet.
- ▶ **rwin=** specifies the size of the TCP receive window for a terminal session. The standard value is 512. The maximum value is 4096. Communication with slow hosts or use of high performance hardware may require a larger TCP receive window.

Chapter 4

AppleTalk

What is AppleTalk?

How AppleTalk works

Seed, nonseed, and soft seed routers

AppleTalk Phase 1/Phase 2

AppleTalk tunnels

Network filtering

Device (NBP) filtering

Kinetics Internet Protocol (KIP)

atalkad

Columbia AppleTalk Package (CAP)

C

C

C

What is AppleTalk?

The AppleTalk network protocols were designed by Apple to let one device communicate with another over LocalTalk, Ethernet, and Token Ring networks. Devices on an AppleTalk network, such as Macintoshes, file servers, and LaserWriters, use AppleTalk protocols to exchange information and services. Apple designed AppleTalk as a low-cost, plug-and-play method of connecting devices to share network resources. Plug-and-play functionality meant that users could connect their Macintoshes to the network and be able to communicate immediately, without complicated configuration and without centralized network administration.

Link Access Protocol (LAP)

The AppleTalk Link Access Protocol (LAP) is responsible for delivery of packets from one node on a network to another node on the same network. The AppleTalk protocol suite uses separate link access protocols for each network medium. The LocalTalk Link Access Protocol (LLAP) controls delivery of packets on a LocalTalk network, and the EtherTalk Link Access Protocol (ELAP) controls delivery on EtherTalk networks. Regardless of the medium, the link access protocols provide similar services to the other AppleTalk protocols.

In addition to providing packet delivery services for a single network, LAP is responsible for assigning network addresses to nodes on an AppleTalk network. For more information on how LAP assigns node addresses, see “Dynamic node address assignment” on page 4-5.

Datagram Delivery Protocol (DDP)

A *socket* is an addressable entity in an AppleTalk node that a client program calls when it needs a network connection. The Datagram Delivery Protocol (DDP) is responsible for opening and closing sockets and for delivery of datagrams from a socket on a node on one network to a socket on a node on another network. Before DDP passes a packet to the appropriate data link for transmission, it adds a short or long DDP header to the packet:

- **Short DDP headers** are used only on unextended networks, where the source and destination sockets have the same network number. A short DDP header lists the datagram length, the source and destination socket numbers, and the DDP type.

- **Long DDP headers** can be used on unextended networks or on extended networks (where the source and destination sockets are on nodes on different networks). A long DDP header lists the datagram length, the complete network/node/socket address of the source and destination sockets, and the DDP type. Long DDP headers can also include a checksum to detect router errors.

Long DDP headers also include a *hop count* field, which the source node sets to 0 when the datagram is sent. Each router that forwards the datagram increments the DDP header hop count. Datagrams with a hop count greater than 15 are discarded to filter out packets trapped in internet loops.

Name Binding Protocol (NBP)

The AppleTalk Name Binding Protocol lets users identify AppleTalk devices by name rather than by network address. NBP is responsible for maintaining maps between device names and their network addresses. Before a device can be accessed over an AppleTalk network or internet, the address for the device must be obtained through the name binding process. See “NBP Lookups (NBPLkUp)” on page 4-6 for a discussion of how name binding works on an AppleTalk network.

Routing Table Maintenance Protocol

The Routing Table Maintenance Protocol (RTMP) lets AppleTalk routers exchange information about the networks to which they are connected and the available routes to remote networks. Every 10 seconds, an AppleTalk router issues an RTMP packet that identifies the networks in its routing table and the distance (in hop counts) from the router to the remote network. A router receiving an RTMP packet compares the networks listed in the packet to the networks in its own routing table, and, if appropriate, adds or updates entries in the routing table to reflect changes in the AppleTalk network topology.

To ensure that routing information remains current, RTMP attaches a timer to each routing table entry. If a router does not see a route to a remote network advertised every 10 seconds, RTMP changes the status of the entry from Good to Suspect. If a suspect route is not advertised within 10 seconds, RTMP changes its status to Bad. If a suspect or bad route is advertised by a router, RTMP changes its status back to Good and restarts its RTMP timer. If a bad route is not advertised within 10 seconds, RTMP

assumes that it is no longer available and deletes the route entry from the routing table.

AppleTalk zones

Apple implemented zones to simplify the interaction between the user and the network. When a router connects two AppleTalk networks, it assigns each network a unique network number. It also specifies the zone to which each network belongs. A zone can consist of one network, or it can group several physical networks into one logical network. LocalTalk networks and Phase 1 EtherTalk networks can only belong to one zone; Phase 2 EtherTalk networks can be associated with as many as 255 zones.

Routing tables

Every AppleTalk router maintains a routing table that describes how datagrams should be forwarded to nodes on remote networks. The routing table contains an entry for each network within 15 hops of the router on the AppleTalk internet. Each routing table entry lists the number of the remote network, the node number of the router through which the network can be reached, and the number of hops (routers) to the network on the AppleTalk internet.

For example, assume that you have an AppleTalk internet made up of four networks connected by AppleTalk routers (Figure 4-1).

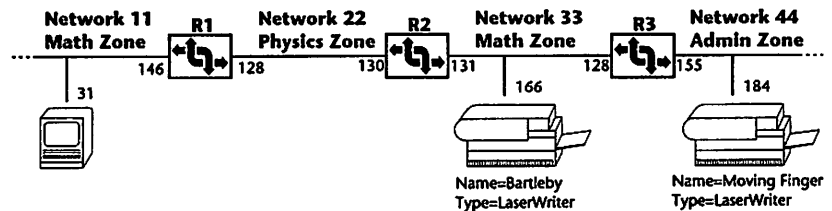


Figure 4-1. AppleTalk routing

Table 4-1 illustrates the routing table for Router R1 in Figure 4-1:

Net	Distance	NextBridge
11	0	-
22	0	-
33	1	22.130
44	2	22.130

Table 4-1. Routing table for router R1

Zone Information Protocol (ZIP)

AppleTalk routers maintain *zone tables* that map network numbers to zone names. The Zone Information Protocol (ZIP) specifies how AppleTalk routers exchange information about the networks to which they are connected and the zones to which those networks belong. When an RTMP broadcast from an AppleTalk router advertises a route to a new network, other routers on the AppleTalk internet send ZIP Queries asking for the zone name associated with the new network. The router receiving a ZIP Query returns a ZIP Reply listing the zone associated with the new network. (In the case of a Phase 2 EtherTalk network, the router can return the list of zones associated with the network.)

If a router wants to send a message to networks in a specified zone, it uses the zone information table to obtain the number of each network. To ensure that the zone table does not include obsolete zone information, ZIP deletes entries from a zone table when they refer to network numbers not listed in the router's routing table.

Table 4-2 illustrates the zone information table for router R1 in Figure 4-1

Net	Zone
11	Math
22	Physics
33	Math
44	English

Table 4-2. Sample Zone Table

How AppleTalk works

Because of the way AppleTalk evolved, it functions differently on a simple AppleTalk network and on an AppleTalk internet (that is, two or more networks connected by a bridge or router).

Simple AppleTalk networks

A simple AppleTalk network consists of a limited number of nodes on a single network cable.

Node addresses

AppleTalk node numbers are eight bits long, meaning that node numbers can range from 0 (00000000) to 255 (11111111):

- ▶ Node number 0 is not used.
- ▶ Node numbers in the range 1-127 are used to identify **client nodes**, such as Macintoshes or personal computers with PC AppleShare.
- ▶ Node numbers in the range 128-254 identifies **server nodes**, such as an AppleShare file server or a LaserWriter.
- ▶ Node number 255 is the AppleTalk broadcast address.

Dynamic node address assignment

Unlike TCP/IP networks, where each node has a fixed network address, AppleTalk assigns network addresses *dynamically*. When a device connected to an AppleTalk network is turned on for the first time, it selects a node number in the appropriate client or server range at random. To ensure that the node number is not already assigned to another node, the device broadcasts an *enquiry control packet* to all nodes on the network. The enquiry control packet asks each node if it is using the selected node number. If the node number has been assigned to another node, the node sends an *acknowledgment control packet* that says "I am using that number." If the first device receives an acknowledgment control packet, it selects another node number from the appropriate range and broadcasts another enquiry control packet. This number selection/testing cycle continues until it finds a node number not claimed by other devices on the network.

Once the device acquires a valid node number, it stores it in its parameter random access memory (PRAM). Thereafter, the device tries to use the stored node number instead of a randomly generated number when it restarts.

AppleTalk transmissions

Like TCP/IP, AppleTalk supports both directed transmissions and broadcast transmissions:

- ▶ A **directed transmission** is a packet sent from a source node to a single destination node. A source node begins a directed transmission by sending a Request to Send (RTS) packet to the destination node. If the destination node is able to accept the connection, it returns a Clear to Send (CTS) packet. Once the source node receives the CTS packet, it begins the actual transmission of data.
- ▶ A **broadcast transmission** is a packet sent from a source node to all nodes on the network. A source node begins a broadcast transmission by sending a Request to Send (RTS) packet to destination address 255. The source node then begins transmission of the broadcast message.

NBP Lookups (NBPLkUp)

Each device on an AppleTalk network is associated with a *name*, a *type*, and a *zonename*. Devices on a simple AppleTalk net locate each other by issuing *NBP Lookup* requests. The syntax for an NBP Lookup is
name : type@zonename.

- ▶ **name** is the name used by the device to identify itself on the network. Macintosh names are entered by means of the Chooser. Other devices, such as GatorBoxes or LaserWriters, are assigned names as part of their initial setup.
- ▶ **type** identifies the kind of device. For example, a LaserWriter (or compatible printer) identifies itself as type `LaserWriter` on the AppleTalk network. Similarly, the NBP type for a GatorBox is `gatorbox`.
- ▶ **zonename** identifies the AppleTalk zone in which the device resides.

Special characters can replace specific values in NBP Lookup requests. For example, assume that the user on the Macintosh (node 31) in Figure 4-2 wants to use a LaserWriter to print a report. The user opens the Chooser and clicks the LaserWriter icon to specify the type of device he wants. The Macintosh broadcasts an NBP Lookup in the format `.../255/31/.../=LaserWriter@*`, where:

- ▷ 255 specifies that the packet is a broadcast
- ▷ 31 specifies the source node for the broadcast
- ▷ = is a wildcard character meaning “any name”
- ▷ LaserWriter specifies the device type
- ▷ * specifies the same zone as the sender

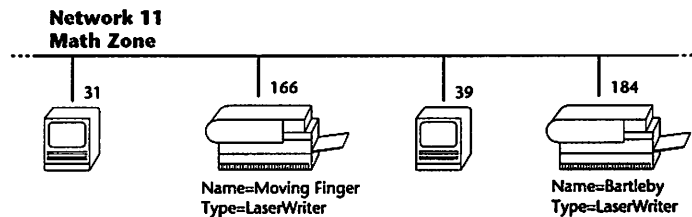


Figure 4-2. Simple AppleTalk network

NBP Replies

Each node on the network receives the NBP Lookup broadcast and compares the request type to its own device type. If the two types match, the node sends an NBP Reply with its name and node number. For example, Bartleby (node 184 in Figure 4-2), would issue an NBP Reply in the format `31/184/.../Bartleby:LaserWriter@*`, where

- ▷ 31 specifies the destination node for the reply (that is, the node address of the device that issued the original NBP Lookup)
- ▷ 184 specifies the node address of the LaserWriter issuing the NBP Reply
- ▷ Bartleby specifies the name assigned to the LaserWriter

- ▷ LaserWriter specifies the device type
- ▷ * specifies the same zone as the sender

The user's Chooser would then list Bartleby as an available LaserWriter. As other devices, such as Moving Finger (node 166) reply, the Chooser adds their names to the list of available LaserWriters.

If the user selects Bartleby, the Macintosh stores the device name in parameter RAM. If a print job is later submitted, the Macintosh issues an NBP Lookup broadcast for Bartleby:LaserWriter@* rather than =:LaserWriter@*.

Complex AppleTalk networks

Joining two AppleTalk networks creates the possibility of nodes on separate networks having the same node number. To ensure that each node on an AppleTalk internet has a unique address, you must assign a separate identification number to each network when you set up a router. When a router is turned on, it establishes itself as a node on each network to which it is connected.

NBP Broadcast Requests (NBPBrRq)

When an AppleTalk node on an AppleTalk Phase 2 internet wants to perform a name-to-address lookup, it issues an NBP Broadcast Request (NBPBrRq) to the router for its network instead of broadcasting an NBP Lookup. Where a node on a simple network issues the NBP Lookup as a broadcast, a node on an AppleTalk internet sends the NBP Broadcast Request as a directed transmission to the nearest router on its own network. If the NBP Broadcast Request only involves the router's local networks, the router issues an NBP Lookup on behalf of the node. If the NBP Broadcast Request involves one or more remote networks, the router converts the NBP Broadcast Request to a Forward Request (FwdReq) and sends the Forward Request to each router connected with the remote networks.

Each destination node will reply to the router from which it received the NBP Lookup; the router in turn forwards each reply to the node that issued the original NBP Broadcast Request.

To illustrate how this works, assume that Node 31 in Figure 4-3 (the Macintosh on Network 11) wants to send a print job to a LaserWriter.

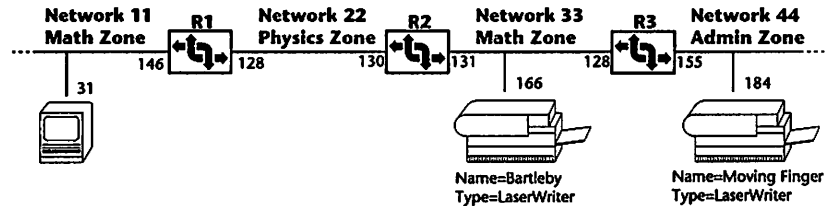


Figure 4-3. AppleTalk internet

When the user opens the Chooser, the following AppleTalk transactions take place. Table 4-3 summarizes the address information for the AppleTalk transactions.

1. The Macintosh issues a ZIP request to the router for its network (R1) asking for all zones in the AppleTalk internet.
2. R1 sends back a list of known zones, which the Macintosh displays in the Chooser window.
3. The user clicks the name of the Math Zone.
4. The user clicks the LaserWriter icon in the Chooser.
5. The Macintosh issues an NBP Broadcast Request for `=:LaserWriter@Math` to R1 (Transaction 1).
6. R1 checks its zone table to identify the numbers of networks in the Math Zone (networks 11 and 33).
7. R1 checks its Routing Table to identify whether it can reach each network in the Math Zone directly or whether it must forward requests for broadcasts to other routers. It learns that network 11 is directly connected and that it must send a request to R2 to reach network 33.
8. R1 issues an NBP Lookup on Network 11 for `=:LaserWriter@*` (Transaction 2).
9. R1 sends an NBPBrRq to R2 for `=:LaserWriter@Math` (Transaction 3).
10. R2 issues an NBP Lookup broadcast (node 255) on network 33 for `=:LaserWriter@*` (Transaction 4).

11. Bartleby (Node 166 on Network 33) sends NBP Reply (Bartleby:LaserWriter@Math) to R2 (Transaction 5).
12. R2 looks up the destination network (11) for the NBP Reply in its routing table and determines that it should forward the reply to R1.
13. R2 forwards the NBP Reply from Bartleby to R1 (Transaction 6).
14. R1 forwards the NBP Reply from Bartleby to the Macintosh (Transaction 7).
15. Bartleby is listed as an available LaserWriter in the Math Zone.

Action	LAP		DDP			
	Destination Node	Source Node	Destination Network	Source Network	Destination Node	Source Node
1. Mac issues NBPBrRq to R1	146	31				
2. R1 issues NBP Lookup to 11	255	146				
3. R1 sends FwdReq to R2	130	128	33	11	255	31
4. R2 issues NBP Lookup	255	131	33	11	255	31
5. Bartleby sends reply to R2	131	166	11	33	31	166
6. R2 forwards reply to R1	128	130	11	33	31	166
7. R1 forwards reply to Mac	31	146	11	33	31	166

Table 4-3. Transaction exchange for looking up LaserWriter

Seed, nonseed, and soft seed routers

Every AppleTalk network on an internet must have a unique number or (in the case of Phase 2 EtherTalk) a unique range of network numbers. Each network must also belong to an AppleTalk zone. AppleTalk provides two ways for a router to identify the number (or number range) and zone (or zone list) associated with a LocalTalk or EtherTalk network: *seed routing* and *nonseed routing*. The GatorBox offers a third option, called *soft seed routing*. Each option is described beginning on page 4-11.

You can independently configure the LocalTalk, Phase 1 EtherTalk, and Phase 2 EtherTalk ports in a GatorBox for seed, nonseed, or soft seed routing. For example, you can specify that your GatorBox should act as a seed router on its LocalTalk network, a nonseed router on its Phase 1 EtherTalk network, and a soft seed router on its Phase 2 EtherTalk network.

Seed router

A *seed router* is configured by the network administrator with the number and zone for each network to which it is connected. When a seed router is turned on, it immediately informs other nodes about its network and zone information by means of RTMP and ZIP packets. It uses its configured information to route AppleTalk packets. If the network information for the seed router conflicts with network information from other routers, the seed router generates a diagnostic message logging the conflict but does not change its network configuration settings. Instead, it continues to broadcast its predefined network information.

Every seed router on a network must be configured with consistent network information. To avoid network information conflicts, the network administrator should turn on seed routing in a limited number of centrally-located GatorBoxes.

Nonseed router

Not every router must have its network number(s) and zone information configured by a network administrator. If a router is set up as a *nonseed router*, it initially assigns its network a network number of 0. It does not inform other nodes about its existence when it is turned on. Instead, a GatorBox configured as a nonseed router will wait two seconds to acquire its network information (network range, zone list, and default zone) from RTMP and ZIP packets generated by another router (*passive discovery*). If the nonseed router does not receive the necessary network information in the first two seconds, it broadcasts RTMP Request or ZIP GetNetInfo Request packets every half second for three seconds to request configuration information from another router on its network (*active discovery*). If it does not receive all necessary network information from one source router by the end of the active discovery period, the GatorBox will not route through the unseeded port.

Note that “nonseed” means that the router does not use preconfigured network information when it starts up; it does *not* mean that it cannot provide network information to other routers after it is running. Once a nonseed router has been “seeded” with network information and begins routing AppleTalk packets, it can seed another nonseed router with its network information when the new nonseed router is turned on.

Once a nonseed router has been “seeded” with network information and begins routing AppleTalk packets, it retains the configuration information it was seeded with. If your network has two seed routers with conflicting network information, the nonseed router will use the information provided by whichever router it hears from first. If the configuration information in the seed router is later changed, the nonseed router must be restarted before it will use the new configuration.

Soft seed router

Soft seed routing combines many of the benefits of seed and nonseed routing. When a GatorBox is set up as a soft seed router on its LocalTalk, Phase 1 EtherTalk, or Phase 2 EtherTalk port, the network administrator enters network number and zone information (exactly as if the GatorBox was a seed router). However, when the GatorBox is turned on, it goes through the same passive and active discovery processes that a nonseed router uses: it uses passive discovery for two seconds to acquire network range, zone list, and default zone information from RTMP packets generated by another source router. If no RTMP packets are received in two seconds, it broadcasts RTMP Request or ZIP GetNetInfo Request packets to request configuration information from any other router. If a GatorBox configured for soft seed routing does not receive network information at the end of the active discovery period, it initializes the appropriate port using the predefined network number and zone information. It then informs other nodes about its network and zone information by means of RTMP and ZIP packets.

When a GatorBox is configured as a soft seed router, its network information will not conflict with the information advertised by other routers operating on the AppleTalk network (if any exist). However, the GatorBox can begin routing if no other router is operating. Soft seed routing is the behavior GatorBoxes running pre-2.0 software releases followed.

AppleTalk Phase 1/Phase 2

The GatorBox software supports both the original AppleTalk protocols (“AppleTalk Phase 1”) and the newer AppleTalk protocols (“AppleTalk Phase 2”). You can specify whether a GatorBox on your internetwork will support AppleTalk Phase 1, AppleTalk Phase 2, or both Phase 1 and Phase 2.

- ▶ You should use only AppleTalk Phase 1 if all your EtherTalk Macintoshes and other EtherTalk devices are running Phase 1.
- ▶ You should use only AppleTalk Phase 2 if you have switched all of your EtherTalk devices to Phase 2.
- ▶ You should use both AppleTalk Phase 1 and AppleTalk Phase 2 if your site is making the transition from Phase 1 to Phase 2. This should be viewed as a short-term solution, since running both Phase 1 and Phase 2 limits how much of Phase 2’s new functionality you can use. When all EtherTalk devices on your internetwork support AppleTalk Phase 2, you can modify the range settings to take advantage of the extended address space and zone lists of AppleTalk Phase 2.

AppleTalk Phase 1

Under AppleTalk Phase 1, each LocalTalk and EtherTalk network on an AppleTalk internet must have a unique network number. The same network number and zone name that you specify for an AppleTalk network must be consistent for all routers on your EtherTalk to identify your Phase 1 EtherTalk network. The network number of your EtherTalk network must be different than the network number of your LocalTalk network, though the two networks can belong to the same AppleTalk zone (if you enter the same zone name for both when you configure your routers).

AppleTalk Phase 2

AppleTalk Phase 2 does not significantly affect the way in which LocalTalk (and LocalTalk-compatible) networks operate. However, AppleTalk Phase 2 changed the way the AppleTalk protocols operate on EtherTalk networks in two ways:

- ▶ **Network addressing** — AppleTalk Phase 2 extends the address space for AppleTalk networks from 254 nodes per network to approximately 16 million nodes. Each AppleTalk Phase 2 network can support a

maximum of 254 nodes, so you must specify a network range broad enough to support all the Macintoshes you plan to place on EtherTalk when you set up the GatorBox. For example, if you have 2000 Macintoshes on EtherTalk and plan to add several hundred more, you could specify a range of 61 to 70, which would support 2540 Macintoshes.

- **Zone lists** — AppleTalk Phase 2 zones still group devices to make it easier to locate and access network services. Under AppleTalk Phase 2, however, a single EtherTalk network can support multiple zones. As a result, you can set up and maintain several zones on a single EtherTalk network.

Each node on the EtherTalk network will initially be assigned to the default zone. Macintosh users on an EtherTalk network can use the Control Panel to select the zone on which they reside. A user's Macintosh will retain its zone settings when it is shut down and will reassociate itself with that zone when it is restarted. If its specified zone is not available when it restarts, the Macintosh will return to the default zone.

The Network range must agree with the range used by all other routers on the Phase 2 EtherTalk network. The network number of your Phase 2 EtherTalk networks must be different than the network number of your LocalTalk network, though the networks can belong to the same AppleTalk zone.

Phase 1/Phase 2 transition

The original AppleTalk was designed to let Macintoshes, LaserWriters, and other devices communicate easily over LocalTalk and Ethernet cabling. Because AppleTalk was originally envisioned as a networking solution for small work groups, it was limited to a maximum of 254 nodes. Under AppleTalk Phase 1, a LocalTalk or EtherTalk network is identified by a unique network number and all devices on a network belong to the same zone.

If Phase 1 and Phase 2 AppleTalk are both turned on, the Phase 2 network will be “advertised” to the Phase 1 network only if the start and end of the Phase 2 range are the same (for example, from 3 to 3). When the GatorBox is set up as a Phase 1/Phase 2 transition router, it translates between Phase 1 messages, such as NBPLookup requests and Phase 2 messages, such as NBPFWdReq requests.

AppleTalk tunnels

AppleTalk tunnels let you route AppleTalk packets from one GatorBox to another over an IP internet by encapsulating the packets inside IP packets. AppleTalk tunneling lets you link AppleTalk networks over an Ethernet backbone without using EtherTalk. This is an advantage for sites that allow only IP protocols on their Ethernet backbone or sites that do not have AppleTalk routing between Ethernet networks.

For example, Figure 4-4 illustrates an IP internet where two Ethernet networks are connected by means of one or more IP routers and GatorBoxes connect LocalTalk networks to the Ethernet networks. If an AppleTalk tunnel is set up between the two GatorBoxes, a Macintosh behind GatorBox1 could see a Macintosh or LaserWriter behind GatorBox2 as if they were communicating over the same AppleTalk network (the tunnel). However, the packets the GatorBoxes exchange are actually encapsulated by one GatorBox, sent across the IP internet as IP packets, and de-encapsulated by the other GatorBox.

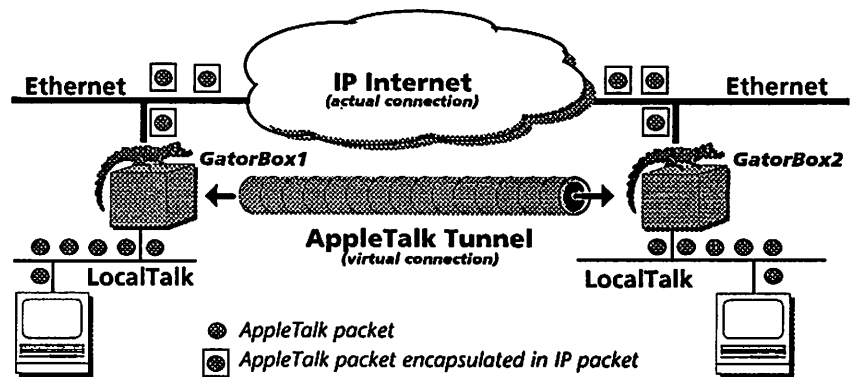


Figure 4-4. AppleTalk tunneling

You can enter up to 32 remote connection points (that is, you can point your GatorBox at 32 other GatorBoxes). Once you connect remote AppleTalk networks together by means of a tunnel, the connected networks function as a single large AppleTalk network. Consequently, you must coordinate network numbers for sites connected with AppleTalk tunnels to avoid number conflicts between networks in different locations.

GatorBoxes connected by AppleTalk tunnels exchange packets as if they are on the same network. In addition to packets traveling from one end node

to another, the GatorBoxes exchange route and zone information packets through the AppleTalk tunnel. If you set up one tunnel to connect GatorBox1 and GatorBox2, and a second tunnel to connect GatorBox2 and GatorBox3, then the three GatorBoxes will typically see and advertise each other's networks and zones. If network filters (described below) are not set up, GatorBoxes in one location can advertise network resources in other locations to all Macintoshes on its network. Users in one location can find their Chooser zone list filled with the names of zones on remote AppleTalk networks.

Network filtering

If you have set up an AppleTalk tunnel (explained above) between two GatorBoxes, you can set up network filters to tell the GatorBox which remote AppleTalk networks should be accessible to users behind the GatorBox. By implementing network filtering, you can restrict the remote networks (and zones) to which users have access.

As a simplified example, assume that an AppleTalk tunnel connects your offices in Boston and San Francisco (Figure 4-5). The Boston office has four zones (networks): (Engineering (network number 11), Marketing (12), Administration (13), and Support (14)). The San Francisco office has one (Sales (21)). Without RTMP filtering, a user in the San Francisco office would see the zones on all five networks. By setting up an RTMP filter to limit access to the Marketing (12) and Admin (13) networks, you can limit the number of Boston zones that appear in the Chooser of the San Francisco user.

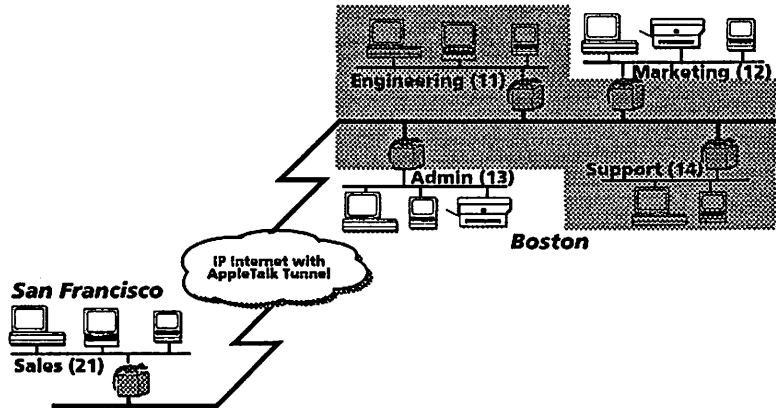


Figure 4-5. Network filtering

The network filters you set up apply to all tunnels running from a GatorBox. Extending the example illustrated in Figure 4-5, assume that you set up a second AppleTalk tunnel to the Sales (11) zone in the Seattle office. If you specify that the San Francisco GatorBox should exclude the Engineering network in Boston (which also has a network number of 11), you would exclude the Seattle network at the same time. Note that this problem could be avoided by coordinating network numbers among the three offices.

The network filters you set up apply to all tunnels, even though the network numbers correspond to nets on the other side of the tunnels. Note that you must coordinate network numbers for sites connected with AppleTalk tunnels to avoid number conflicts between networks in different locations.

Device (NBP) filtering

When you use the Chooser to select a network resource, such as a file server or a printer, your Macintosh issues an NBP Lookup to identify what devices are available. For example, if Nancy (in the Sales zone) clicks the LaserWriter icon in the Chooser, her Macintosh issues an NBP Lookup in the format `=:LaserWriter@Sales`, which means "What LaserWriters are available in the Sales zone?" Any LaserWriter in the specified zone would respond with its name, letting Nancy select a printer or look in another zone if she doesn't see the printer she wants.

By setting up the GatorBox to filter NBP transactions, you can restrict what information about network resources appears in users' Choosers. Your GatorBox monitors NBP Lookup messages and replies or forwards only those that meet the filter criteria you specify. You can set up three types of device filters:

- ▶ Stay-in-zone filtering
- ▶ Laser filtering
- ▶ Device name (tilde) filtering

Each type of device filter is described below.



Unlike network filters, which are only used in conjunction with AppleTalk tunnels, NBP filters apply to local AppleTalk networks and remote (tunnel) AppleTalk networks. You do not need to set up an AppleTalk tunnel to use NBP filtering.

Stay-in-zone filtering

Stay-in-zone filtering prevents Macintosh users on the LocalTalk network behind a GatorBox from seeing AppleTalk devices in other zones. When you turn on stay-in-zone filtering, your GatorBox drops NBP Lookups to zones other than the origination LocalTalk zone. As a result, users in a filtered zone cannot see zones or devices on the other side of the GatorBox and users outside the filtered zone cannot see devices within the zone.

For example, assume that the Sales department has an AppleTalk network connected to Ethernet via a GatorBox and that the zone name is Sales (Figure 4-6). When you turn on stay-in-zone filtering in GatorBox 1, Nancy (in the Sales zone) can only see LaserWriters, NetModems, and other devices that are also in the Sales zone. The Engineering zone will not appear in her Chooser window, so she won't see devices, such as Harpo or Groucho, in the Engineering zone.

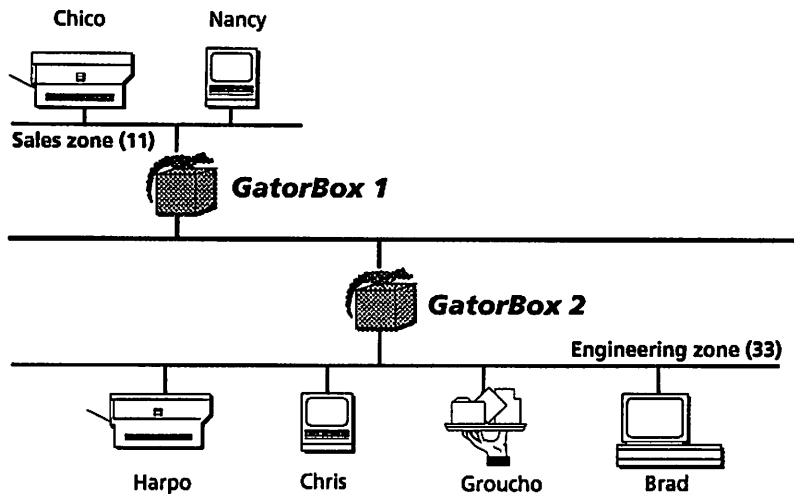


Figure 4-6. Stay-in-zone filtering

Stay-in-zone filtering works on the basis of *zone name*. If you have two AppleTalk networks in the same zone (Sales zone in Figure 4-7), a user on one network will see devices on the other network. For example, if stay-in-zone filtering was turned on for GatorBox 1, Nancy could access Zeppo (the AppleShare file server) as well as Chico (the LaserWriter), but could not see Groucho or Harpo in the Engineering zone.

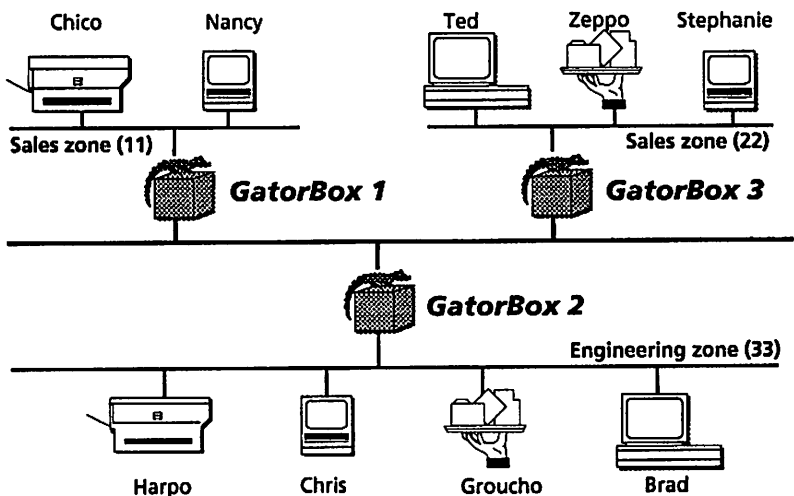


Figure 4-7. Filtering between networks with the same zone name

Stay-in-zone filtering can be implemented on a GatorBox-by-GatorBox basis. For example, assume you turn on stay-in-zone filtering for GatorBox 1 but leave it off for GatorBox 2 and GatorBox 3:

- ▶ Nancy (in the Sales zone connected to GatorBox 1) will only see devices, such as Chico and Zeppo, in the Sales zone.
- ▶ Chris (in the Engineering zone) will be able to see Harpo, Groucho, Zeppo, and Chico.
- ▶ Ted (in the Sales zone connected to GatorBox 3) will be able to see Groucho, Harpo, Chico, and Zeppo.

Laser filtering

Where stay-in-zone filtering shields all devices in a zone, laser filtering lets you shield LaserWriters behind the GatorBox from being seen by anyone outside its AppleTalk zone. When laser filtering is implemented, the GatorBox monitors replies to NBP requests. When the source and destination zones for the reply do not match and the device type is `laserwriter`, the GatorBox does not forward the reply. Consequently, users in a filtered zone cannot see LaserWriters outside their zone and users in other zones cannot see or use a LaserWriter in a shielded zone.

For example, assume that stay-in-zone filtering has been turned off and laser filtering has been turned on for all three GatorBoxes in Figure 4-8. Ted and Nancy can see Chico (the LaserWriter in the Sales zone) but not Harpo (the LaserWriter in the Engineering zone). Chris can see Harpo (the LaserWriter in the Engineering zone) but not Chico (the LaserWriter in the Sales zone). However, everyone can access Zeppo and Groucho, the AppleShare servers in the two zones.

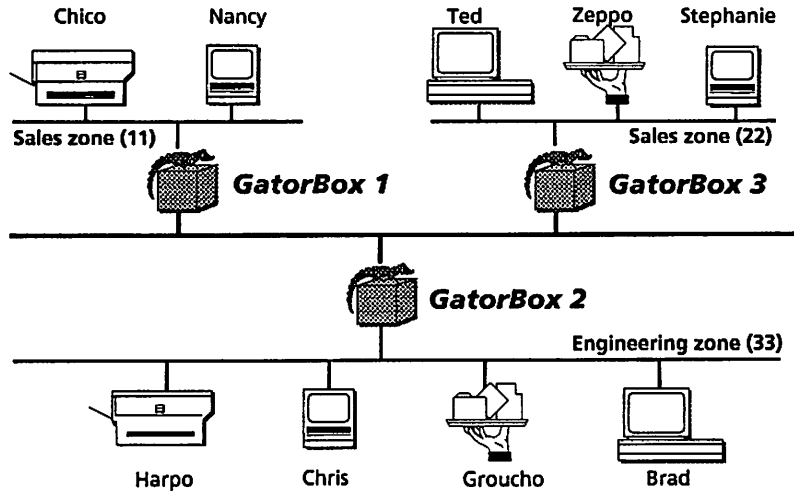


Figure 4-8. Laser filtering

Device name (tilde) filtering

Device name filtering allows you to shield any device that has a tilde (~) character at the end of its name from being seen outside its zone. For example, in Figure 4-9, you could change name of the AppleShare server in the Engineering zone from “Groucho” to “Groucho~” and turn on the tilde filter for GatorBox 2 to prevent access by users in the Sales zone.

Device name filtering will not work when the GatorBox’s LocalTalk network is connected to another LocalTalk network by means of a LocalTalk router, such as a Hayes InterBridge. Devices on the other side of the LocalTalk router will be able to see all devices on the GatorBox side of the bridge.

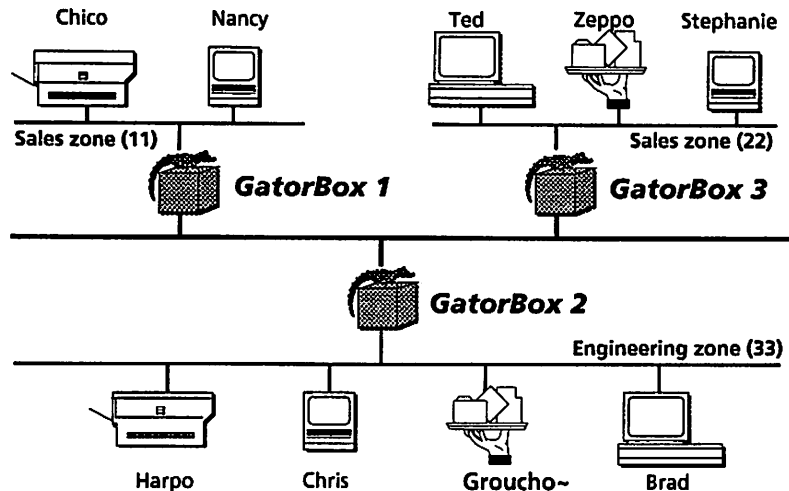


Figure 4-9. Device name (tilde) filtering

Kinetics Internet Protocol (KIP)

The GatorBox supports the Kinetics Internet Protocol (KIP) (also referred to as IPTalk), which encapsulates AppleTalk packets inside UDP/IP packets. UDP/IP encapsulation lets a Macintosh on LocalTalk or EtherTalk access IP-based computers that understand AppleTalk protocols. When KIP support is enabled, the GatorBox can support the Columbia AppleTalk Package (CAP) software (described on page 4-24).

Refer to the *GatorBox User's Guide* for information about how to set up KIP.

UDP port range

Early releases of KIP used a range of UDP ports, starting at 768, to map to the “well-known” DDP sockets. More recent releases of KIP use a range of ports assigned by the Network Information Center that begin at port 200. You can specify that UDP port range the GatorBox is to use by clicking the **Use New UDP Port Range (200)** checkbox in the KIP Options dialog box.

If you specify that you want use the new UDP port range, you must add the following lines to the `/etc/services` file on your CAP server:

```
at-rtmp  201/udp  # AppleTalk Routing Maintenance
at-nbp   202/udp  # AppleTalk Name Binding
at-echo  204/udp  # AppleTalk Echo
at-zis   206/udp  # AppleTalk Zone Information
```

atalkad

The atalkad (AppleTalk administration daemon) software was developed at Stanford University to provide centralized administration of AppleTalk networks. atalkad runs on a UNIX host on your internet and answers requests for configuration information from AppleTalk routers. atalkad can use AppleTalk routing tables maintained on a UNIX machine to download some of the GatorBox's configuration information. atalkad can also be used to propagate routing information to routers on your AppleTalk internet.

Although using atalkad lets you link AppleTalk networks across IP networks, atalkad requires the use of a UNIX system to configure the GatorBox and will not work in a Phase 2 environment. atalkad support will be most useful to sites that prefer centralized network administration and that do not anticipate converting to AppleTalk Phase 2.

atalkatab

The `/etc/atalkatab` (AppleTalk administration database table) file holds the configuration settings used by atalkad to configure the AppleTalk routers on your network. You must update the `/etc/atalkatab` file with information about your AppleTalk internet.

Refer to the documentation supplied with atalkad for information about setting up your `/etc/atalkatab` file.

/etc/ataalk.local

Each UNIX host that will use CAP utilities and libraries must have its own `/etc/ataalk.local` file, which lists the AppleTalk address of the UNIX host and the AppleTalk address of the GatorBox (or other gateway). If you intend to use the GatorBox's KIP functionality to support CAP, you need to provide GatorKeeper with the CAP parameters specified in the `/etc/ataalk.local` file on one of your CAP hosts.

Each UNIX host that implements CAP must have a `/etc/ataalk.local` file. The format for the `/etc/ataalk.local` file is:

```
#mynet      mynode      myzone
<net>       <node>       <zonenumber>

#bridgenetbridgenode  bridgeIP
<net>       <node>       <zonenumber>
```

where:

- ▶ `mynet` is the network number that identifies the KIP network.
- ▶ `mynode` is the low byte of the CAP server's IP address.
- ▶ `myzone` is the name of the AppleTalk zone in which the CAP server resides.
- ▶ `bridgenet` is the IP address of the closest GatorBox (or other gateway).
- ▶ `bridgenode` is the low byte of the GatorBox's IP address.
- ▶ `bridgezone` is the name of the AppleTalk zone in which the GatorBox resides.

Columbia AppleTalk Package (CAP)

The Columbia AppleTalk Package (CAP) is a set of daemons that run on a UNIX host. CAP encapsulates AppleTalk packets inside IP datagrams for transmittal over an IP network. CAP provides limited file sharing and bidirectional (UNIX-to-LocalTalk and LocalTalk-to-UNIX) print spooling capabilities.

Chapter 5

DECnet

What is DECnet?

What is a DECnet router?

Hello messages

Router messages

C

C

C

What is DECnet?

DECnet is a networking architecture designed by Digital Equipment Corporation that lets DIGITAL computers and operating systems communicate with each other. DECnet services include terminal services, remote file access, and electronic mail.

Areas and nodes

The network administrator can partition a large DECnet internet into *areas*, where an area can represent a single network or a group of networks. DECnet areas are analogous to IP and AppleTalk networks.

Each device on a DECnet network that has its own address is a *node*. Each node must belong to one (and only one) DECnet area.

DECnet addresses

A DECnet node is identified by a 16-bit number that identifies its *area number* and its *node number*.

- ▶ The **area number** is a number, in the range 1-63, that designates the area in which the node is grouped. DECnet areas are analogous to IP and AppleTalk networks.
- ▶ The **node number** is a number, in the range 1-1023, that designates the node's unique address within the area.

A DECnet address is written in the format `area_number.node_number`. Unlike TCP/IP and AppleTalk, DECnet does not differentiate between physical addresses and logical addresses. A DECnet node actually changes its physical Ethernet address when its node or area number changes. For this reason, a GatorBox may change its Ethernet address when it is configured for DECnet routing.

What is a DECnet router?

A DECnet router is a network node that passes DECnet packets from one network to another. DECnet routers belong to one of two classes:

- ▶ **A DECnet level 1 router** keeps track of nodes in its own area and the closest level 2 router in its area. A level 1 router passes packets from one network to another within a single DECnet area. If a packet is addressed to a node in another area, the level 1 router must forward the packet to its level 2 router.
- ▶ **A DECnet level 2 router** keeps track of the paths to destination areas and passes packets from one area to another.

The GatorBox meets Digital's specifications for a level 1 router. The GatorBox acts as a gateway to DECnet for Macintosh computers using the AppleTalk protocol. Macintosh nodes on AppleTalk encapsulate DECnet packets inside AppleTalk packets and use the gateway to communicate with DECnet nodes on Ethernet. Once a Macintosh is connected to a DECnet node, it can exchange mail, access and manipulate files on the DECnet nodes, and use terminal services to connect to other computers from separate Macintosh windows.

How does DECnet routing work?

Nodes on AppleTalk identify a DECnet router by means of AppleTalk Name Binding Protocol (NBP) Lookups. These nodes then communicate with the router using a simple DDP protocol that encapsulates the DECnet packets. Using an application such as Digital Equipment Corporation's PATHWORKS, the Macintosh user sends a stream of packets to the DECnet router.

When the DECnet router receives a packet, it verifies that the packet is of an acceptable size. The DECnet router then parses the DECnet header and obtains the source area and node, the destination area and node, and the hop count (that is, the number of routers through which the packet has traveled). The router compares information in the packet header to information in its DECnet routing table to identify the most efficient path for delivering the packet.

- ▶ If the destination node is on a directly connected network in the same DECnet area, the router will transfer the packet to the destination node.

- ▶ If the destination node is on another network in the same DECnet area, the router will identify the most efficient path for routing the packet. It will then forward the packet to the next router on the selected path.
- ▶ If the destination node is on a network in another area, the router will deliver the packet to the closest DECnet level 2 router. The level 2 router will then assume responsibility for routing the packet to a level 2 router in the correct area, which will, in turn, direct the packet to the destination node or a level 1 router in that area.

Routing tables

As part of its operations, a DECnet router must process and store information about changes in the topology of its Ethernet network and connected networks. The GatorBox stores this information in a routing table. Each entry in the routing table represents a reachable end node in the router's area. The routing table entry for each node includes the following:

- ▶ **DECnet address** — The address of a node in `area.node` format
- ▶ **Route to the given address** — The address of the router or gateway through which a packet must pass to reach the node. Depending on the network involved, this can be either a DECnet address or an AppleTalk address.
- ▶ **HELLO timer** — The frequency with which the node is expected to issue HELLO packets. If a router doesn't hear from a node in three HELLO cycles, it assumes that the node is no longer reachable.
- ▶ **Cost** — The factor the router uses to select the most efficient (least costly) route to a destination node.
- ▶ **Hop count** — The number of routers through which a packet must pass to reach the destination node.
- ▶ **Maximum segment size** — The maximum number of bytes that can be included in a datagram to the node.

The first entry in the routing table (entry 0) is reserved for the nearest level 2 router.

Minimum cost calculation

A DECnet router will try to select the most efficient path for routing a packet (that is, it will select a route to a destination node that minimizes the cost of the packet transmission). Because some paths use slower media than others, the most efficient path is not necessarily the one with the lowest hop count. Instead, a network administrator assigns each leg of a path a number indicating its cost. The cost of a path is therefore the sum of the costs associated with each leg of the path.

For example, assume that Node 1 in Figure 5-1 wants to send a packet to Node 6.

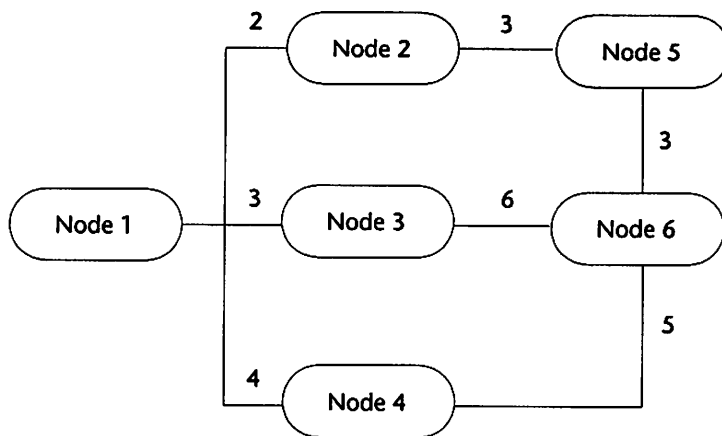


Figure 5-1. Path cost diagram

As Table 5-1 indicates, Node 1 has three paths available to reach Node 6:

Path	Hop count	Cost
1-3-6	1	9 (3+6)
1-2-5-6	2	8 (2+3+3)
1-4-6	1	9 (4+5)

Table 5-1. Path cost comparisons

Even though Path 1-2-5-6 has a greater hop count, it has the lowest cost associated with it. Node 1 would therefore determine that 1-2-5-6 is the most efficient route.

Hello messages

Each node on a DECnet network periodically sends a Hello message to every router in its area, informing the router that it is still accessible. The suggested period for sending Hello messages is 30 seconds. The router maintains a Hello timer for each end node in its routing table. If a router does not receive a Hello message for three successive periods from an end node, it assumes that the end node is no longer available, and deletes the node entry from its routing table.

Similarly, every router multicasts a Hello message to every end node, informing the node that it is still available to route packets. If an end node does not receive a Hello message for three successive periods from a router, it assumes that the router is no longer available. End nodes on LocalTalk will then use NBPLookups to discover a new DECnet router (if any) on its network.

Router messages

Every level 1 router on a DECnet network periodically multicasts routing messages to every other router in its area. Each routing message informs the other routers of the contents of its routing table, including node numbers, path lengths (hop counts), and path costs.

C

C

C

Chapter 6

UNIX-to-LocalTalk Printing

About **lpr**

About **PAP**

How GatorPrint works

PostScript translation

International character mapping

/etc/printcap file

UNIX operating systems derived from the Berkeley Software Distribution (BSD) version, such as SunOS or ULTRIX, include a line printer protocol called `lpr` (line printer remote). `lpr` lets users submit print requests to a queue on a local or remote host, which handles spooling and printing the job.

The Macintosh operating system uses the AppleTalk PAP (Printer Access Protocol) to manage the interaction between print *clients* (typically Macintoshes) and printer *servers* (typically AppleTalk printers). PAP handles connection setup, maintenance and termination, as well as data transfer between the print client and server.

GatorPrint bridges the `lpr` and PAP printing worlds by making a GatorBox look like a remote UNIX host to UNIX print clients and like a PAP print client to LocalTalk printers. When a UNIX user submits a print request that specifies a LocalTalk printer, your UNIX host forwards the print job to the GatorBox using the `lpr` protocol. The GatorBox passes this information to the AppleTalk printer using the PAP protocol.

About lpr

BSD UNIX users use the `lpr` command to access printers. The `lpr` command calls the line print daemon, `lpd`, to print files from a queue, transfer files to the spooling area, display print queues, or remove jobs from a queue.

When you submit a print request on a UNIX system by issuing an `lpr` command, the `lpd` accepts the request and puts a print job into a spool directory on the local machine. Two types of files are created for each print job:

- ▶ The **data file** for a print job contains the information (text and graphics) that is actually printed. If you specify that more than one file should be printed as part of a print job, a separate data file is created for each source file.
- ▶ The **control file** for a print job specifies the file or files to be printed and the non-printing actions to be performed. The control file identifies the name of the user who submitted the job, the size of the print job, and other information.

The `printcap` file, which is typically located in the `/etc` directory on a UNIX host, is a database that describes the printers that can be accessed through a direct connection or over a network. The `/etc/printcap` file specifies a spool directory for each printer. The `/etc/printcap` file is described in “`/etc/printcap` file” on page 6-12.

If you specify a remote printer (that is, a printer not connected directly to your UNIX host) when you issue the `lpr` command, `lpd` establishes a connection with the remote UNIX host to which the printer is connected and sends the print job to it.

Figure 6-1 identifies the steps that a UNIX system follows when a print job (“MyReport”) is submitted from UNIX host (“Fred”) to a remote UNIX printer (“Pebbles”).

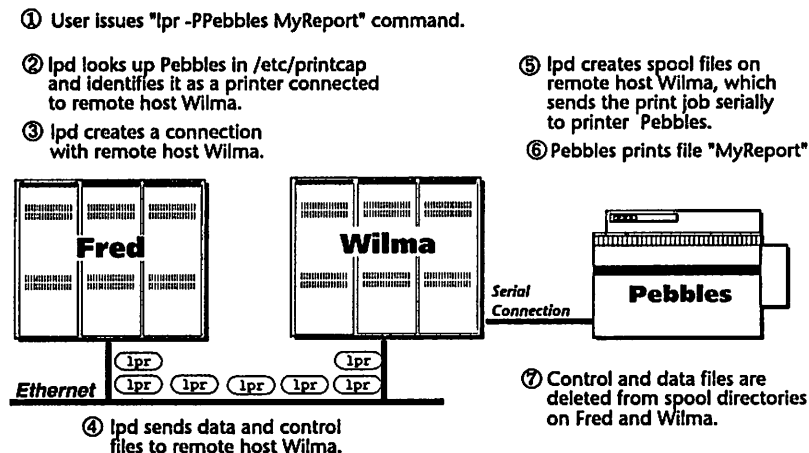


Figure 6-1. lpr printing

Operating systems that support lpr

Your UNIX system must support or simulate the `lpr` protocol for GatorPrint to function properly. Among the UNIX operating systems that have been tested for compatibility with GatorPrint are:

- ▶ A/UX 2.0 (Apple)
- ▶ Apollo's OS with/ DOMAIN/IX & set up with/ Berkeley derivatives

- ▶ Convex V7.0
- ▶ DEC BSD 2.10.1, 4.3
- ▶ HP UNIX 6.5, 7.0
- ▶ HP9000
- ▶ MASSCOMP 5600
- ▶ MIPS OS 4.3
- ▶ MS/DOS 4.01 with NCSA Telnet/lpr 2.3b9
- ▶ NeXT OS
- ▶ Pyramid MIS-1/1LC — OSx16NQ System V BSD
- ▶ Pyramid OSx 5.0b
- ▶ Sequent Dynex 3.0.17
- ▶ Silicon Graphics Iris — OS 3.3
- ▶ Sony NeWS OS
- ▶ SUN OS
- ▶ Tektronix 4315, 4317, 4301
- ▶ ULTRIX v3.0, v4.0
- ▶ VMS with lpr extensions

If you are not sure whether your UNIX system supports lpr, consult your system administrator or your UNIX documentation and/or vendor.



If your system is running a version of System V UNIX that does not support lpr, you can obtain a copy of Cayman's lprclient utility from Cayman Technical Services. The lprclient utility is an unsupported program that lets machines running System V lp simulate lpr commands. The lprclient utility, which can be obtained by anonymous ftp from <ftp.cayman.com>, is distributed as a shar (shell archive) file that you unpack, compile, and install on your System V UNIX hosts. The README file for lprclient includes installation instructions and examples.

About PAP

The Macintosh operating system uses the AppleTalk PAP (Printer Access Protocol) to manage the interaction between print *clients* (typically Macintoshes) and printer *servers* (typically AppleTalk printers such as LaserWriters and networked ImageWriters). PAP handles connection setup, maintenance and termination, as well as data transfer between the print client and server.

When a printer is turned on, it registers its name (for example, "Franklin") and printer type (for example, "ImageWriter" or "LaserWriter") on the AppleTalk network. You select a printer from the Chooser by specifying a printer type (and, if necessary, an AppleTalk zone), and then clicking the name of the printer you want to use.

When you submit a print job, the Macintosh issues an NBP (Name Binding Protocol) Lookup to verify that the selected printer is available and to obtain the AppleTalk address of the printer. If the printer is ready to accept a new connection, PAP opens a connection between the Macintosh and printer. Once the connection is open, the Macintosh and printer exchange messages and data. The client Macintosh assigns a sequence number to each packet that is part of the print job. PAP uses this sequence number to identify duplicate packets.

When PAP opens a connection, it starts a connection timer to signal that the connection may have closed on one end or the other. PAP restarts the connection timer whenever a packet is sent between the Macintosh and the printer. Either end of the connection can send tickle packets to indicate that the connection is still open. If two minutes go by without either end of the connection sending a packet, PAP closes the connection.

Since the printer-to-Macintosh connection is dynamic, AppleTalk print clients do not require a static configuration table analogous to the UNIX `/etc/printcap` file to access a printer. This means that an administrator typically cannot restrict access, invoke printer administration, or alter spooling methods without interacting with a local Macintosh application.

Unlike `lpr`, PAP does not use a print queue. Rather, the arbitration of a printer is managed by both the PAP client and the printer. If a client submits a print request to a busy printer, the printer refuses the connection. The client then resubmits the request every two seconds until the request is cancelled. The client updates the print request each time it resends it with

the length of time it has been waiting to submit a job. When the printer becomes available, it polls the network for print requests for two seconds and then accepts the request with the longest wait time.

How GatorPrint works

GatorPrint bridges the lpr and PAP printing worlds by making a GatorBox look like a remote UNIX host to UNIX print clients and like a PAP print client to LocalTalk printers. When a UNIX user submits a print request that specifies a LocalTalk printer, the UNIX host forwards the print job to the GatorBox using the lpr protocol. The GatorBox passes this information to the AppleTalk printer using the PAP protocol. Depending on the format of the file and the destination printer type, the GatorBox may convert the print job:

- ▶ **Text-to-PostScript conversion** — If the print job is in text format and the destination printer uses PostScript, the GatorBox converts the text to a format acceptable to the destination printer. The GatorBox then forwards the converted data to the appropriate AppleTalk printer.
- ▶ **Pre-processed PostScript** — If the print job is already in PostScript format and the GatorPrint PostScript filter is turned on, the GatorBox forwards the PostScript data to the destination printer without converting it.
- ▶ **Non-PostScript printer**— If the destination printer does not use PostScript, the GatorBox forwards the text data to the printer without conversion.

When the job is printed successfully, the GatorBox sends a confirmation message to the UNIX system, informing lpd that the spool files can be deleted.

Figure 6-2 illustrates how a UNIX computer (“Fred”) on an Ethernet network sends a print job through a GatorBox (“GatorPrinter”) to a LaserWriter (“Franklin”) on LocalTalk.

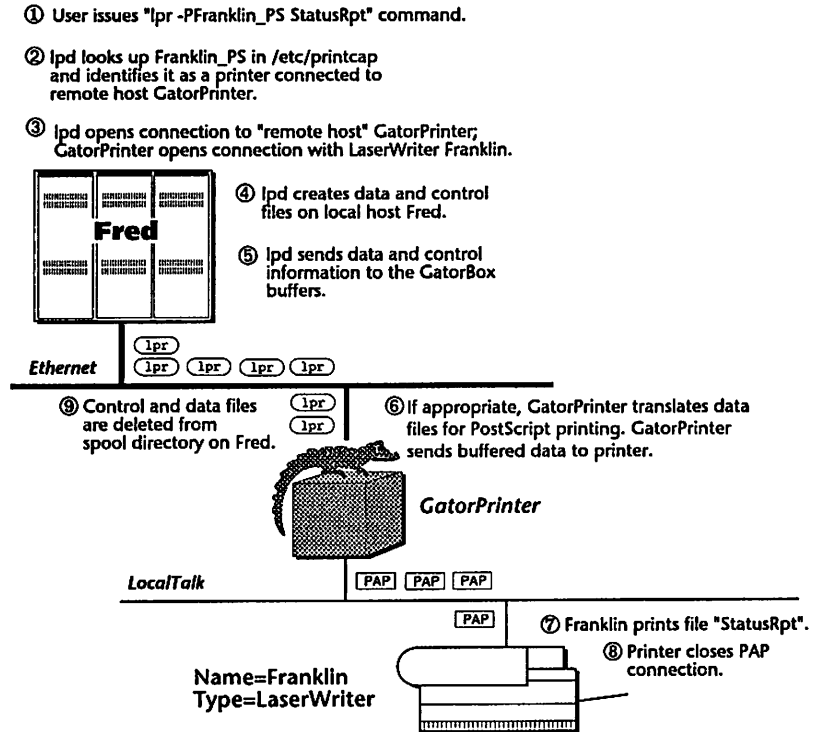


Figure 6-2. lpr-to-PAP printing

Physical and logical printers

A **physical printer** is a device, such as a LaserWriter, that accepts print commands and data from the network and generates text or graphics on paper. For example, Figure 6-2 shows one physical printer ("Franklin") connected to the AppleTalk network.

A **logical printer** is a printer *definition* that specifies how text should be printed by a physical printer. You can create more than one logical printer for a single physical printer. For example, you could create one definition ("Franklin_PS") to print files already in PostScript format, and a second definition ("Franklin_H") to convert text files to a 10-point Helvetica typeface before printing. The GatorBox would regard the two definitions as separate printers, even though they use the same physical printer.

Connection attempts

The GatorBox accepts three connection attempts (or two connection attempts and one connection) for a destination (physical) printer. If a fourth print job is sent through the GatorBox, GatorPrint tells the UNIX lpd that it cannot handle any more jobs. The lpd will wait four minutes before submitting more requests. This lets the printer complete the print jobs already in its queue. Although the four-minute backoff will usually be transparent to users, a high volume of print requests can result in a job being postponed more than once.

Multiple printers and print queues

GatorPrint can process print jobs for as many as six printers simultaneously. Because a print job can be quite large, and because a UNIX system can send data at higher speeds than an AppleTalk device can accept, the GatorBox buffers each print job in segments. Consequently, the GatorBox forwards data to an AppleTalk printer at the same time it receives data from the UNIX system. This differs from lpr-to-lpr print jobs, where the entire file is transferred to the remote host and then sent serially to the printer.

If more than one printer is connected to the LocalTalk network, a GatorBox running GatorPrint can process print jobs for multiple printers simultaneously. If more than one job is waiting to be printed to a physical printer through a GatorBox (or if the GatorBox is turned off), the lpd queues print jobs for later processing. The local lpd has an algorithm for resubmitting the print job so that the job can be printed when the printer is available.

If a user submits a job destined for a printer that is busy, the job is entered in the print queue. When the lpd resubmits the job, the GatorBox again attempts to establish a connection to the printer — this cycle is repeated until the printer becomes available and the job is printed, or until the job is removed from the queue on the UNIX host.

The GatorBox can only accept one connection for each logical printer. If you are sending print jobs from several UNIX hosts, you create unique logical printer definitions for each host. For example, you can specify that Host A will use logical printer QUEUE1 to send print jobs through GatorPrint while Host B will use logical printer QUEUE2. While the two logical printers could route print jobs to the same physical printer, creating separate queues for each host reduces contention for the printer. You can create as many as

32 logical printer definitions for each GatorBox. For examples and more detailed discussion of how to create logical printers, see “/etc/printcap file” on page 6-12.

Printer name

Each printer in an AppleTalk zone must have a unique name. The printer's name, which appears in the Chooser, identifies it to other devices on the AppleTalk network.

Older versions of Apple's Namer utility, which assigns names to LaserWriters, added a blank character to the end of device names with an odd number of characters. If you have difficulties connecting to a printer, add a space at the end of the LaserWriter name you enter in GatorKeeper to test whether the name the LaserWriter is registering on your AppleTalk networks includes a trailing space.

Although AppleTalk printer names can be up to 32 characters long, GatorPrint has problems working with printer names longer than 27 characters.

Printer type

Each AppleTalk printer identifies itself on the AppleTalk network by type as well as by name. The manner in which commands and data are sent to a printer depends on the printer type.

- ▶ **If the printer is a LaserWriter** (or LaserWriter-compatible), the Macintosh uses the PostScript page description language. PostScript is an industry-standard device-independent programming language and print file format.
- ▶ **If the printer is an ImageWriter**, the Macintosh translates the QuickDraw graphics routines to a bit-mapped image. QuickDraw is a graphics model built into the Macintosh ROM to draw images, such as characters or graphics, on the Macintosh screen.

Compatible printers

Among the PostScript printers that have been tested for compatibility with GatorPrint are:

- ▶ Apple LaserWriter Plus/IINT/IINTX
- ▶ Apple Personal LaserWriter IINT
- ▶ GCC BLP
- ▶ HP LaserJet IIID
- ▶ NEC ColorMate
- ▶ NEC SilentWriter LC 890
- ▶ Phoenix
- ▶ QMS ColorScript Model 10
- ▶ QMS PS-800+
- ▶ QMS PS-820
- ▶ VariTyper VT600P/VT600W

GatorPrint has also been tested in environments where a GatorBox and LaserWriter are in the same AppleTalk zone but the devices are separated by a Farallon Star Controller. GatorPrint also supports the AppleShare LaserSpooler software.

PostScript translation

PostScript translation utilities specify the manner in which text should be converted to a format acceptable to PostScript printers. You can configure a GatorBox to convert a text file into a specified type size and type face in portrait (tall) or landscape (wide) format. When a text file is submitted to the GatorPrint PostScript translator, the software inserts the appropriate PostScript commands.

As an alternative to using GatorPrint's text-to-PostScript filters, you can use commercial or public domain PostScript filters, such as Adobe Systems' TRANSCRIPT package, to convert UNIX documents and graphics files to PostScript format before you print the files using GatorPrint. These third-party PostScript filters allow you to convert UNIX files in non-text format, such as TeX or troff, to PostScript before sending them to an AppleTalk printer. If you want to apply a third-party PostScript filter to a print job, you must use the command

```
<mypostscriptfilter>$*|lpr -P<printername>
```

For more information on third-party PostScript filters, consult your hardware vendor.

International character mapping

Some international UNIX operating systems can generate characters based on codes defined in the ISO 8859-1 standard. If you want the ISO international character set supported, you can turn on international character mapping for a logical printer. GatorPrint will filter specially formatted character strings and convert them to the appropriate international characters. For example, GatorPrint will convert the hexadecimal code 6c to the character "Æ" when international character mapping is turned on.

Although you can turn on international character mapping without turning on the GatorBox's text-to-PostScript filter, you should use text-to-PostScript conversion with international character mapping whenever possible. Text converted to PostScript on the host system rather than in the GatorBox may modify the standard PostScript code assignments, limiting the effectiveness of international character mapping in the GatorBox.

Table 6-1 presents the international characters that can be generated by each hexadecimal code when the PostScript conversion is performed by the GatorPrint software. Note that some characters in the ISO 8859-1 specification do not have PostScript equivalents. An international character that does not have a PostScript equivalent will be mapped to the closest PostScript character.

Table 6-2 presents the international characters that can be generated by each hexadecimal code when the PostScript conversion is performed by the UNIX host instead of by the GatorPrint software.

hex	a	b	c	d	e	f
0		°	À	D	à	ð
1	¡	±	Á	Ñ	á	ñ
2	¢	²	Â	Ò	â	ò
3	£	³	Ã	Ó	ã	ó
4	¤	´	Ä	Ô	ä	ô
5	¥	µ	Å	Õ	å	õ
6	¦	¶	Æ	Ö	æ	ö
7	§	•	Ç	×	ç	+
8	¨	¸	È	Ø	è	ø
9	©	¹	É	Ù	é	ù
a	ª	º	Ê	Ú	ê	ú
b	«	»	Ë	Û	ë	û
c	¬	¼	Ì	Ü	ì	ü
d	...	½	Í	Ý	í	ý
e	®	¾	Î	Þ	î	þ
f	¯	¿	Ï	ß	ï	ÿ

Table 6-1. International character mapping with GatorPrint PostScript conversion

hex	a	b	c	d	e	f
0		°	À	Ð	à	
1	¡	+_	Á	Ñ	á	ñ
2	¢	‡	Â	Ó	â	ó
3	£	††	Ã	Ô	ã	ô
4	¤	˙	Ä	Õ	ä	ö
5	¥	u	Å	Ö	å	ø
6	¦	¶	Æ	Ø	æ	ø
7	§	•	Ç	×	ç	/
8	¨		È	Ø	è	ø
9	©	†	É	Ù	é	ú
a	ª	º	Ê	Ú	ê	û
b	«	»	Ë	Û	ë	ü
c		¼	Ì	Ü	ì	ü
d	...	½	Í	Ý	í	ý
e	¸	¾	Î	Þ	î	þ
f	-	¿	Ï	ß	ï	ÿ

Table 6-2. International character mapping without PostScript conversion

/etc/printcap file

The `/etc/printcap` file is a database listing printers that can be accessed through a direct connection or over a network. The `lpd` spooling system re-reads the `/etc/printcap` file every time a print job is submitted. Consequently, you can add and delete printers from the `printcap` file without restarting devices or processes on your UNIX system.

You must add an entry in the `printcap` file for **each logical printer** that you want to access from UNIX. A `printcap` file entry for an AppleTalk printer is set up as if the GatorBox were the remote host and the AppleTalk printer is connected directly to it.



NeXT users can use the PrintManager tool in the NetInfoManager application to set up printers in the NetInfo database instead of using a printcap file. Refer to the appropriate NeXT documentation for information about using NetInfoManager.

For example, you might add the following entry to the printcap file on a UNIX host:

```
#This entry is for printer in Figure 6-2
AT_Printer|Franklin_PS:\
:lp=:rm=GatorPrinter:\
:sd=/usr/spool/lpd/GatorPrinter/Franklin_PS:\
:lf=/usr/adm/lpd-errs:\
:rp=Franklin_PS:
```

The first field in a printcap entry specifies the name (or names) by which a printer can be identified in the `lpr -P<printrname> <filename>` command. The first field in a printcap file must begin at the left margin without a leading colon. If the same printer can be called by more than one name, each alias is separated by a vertical bar (|). For example, `AT_Printer|Franklin_PS` indicates that this printcap entry would be used when a print job specifies `AT_Printer` or `Franklin_PS`.

- ▶ `#` (pound sign) indicates the start of a comment. A comment can begin on its own line or can follow fields in a data line.
- ▶ `:` (colon) separates fields within the printcap entry. If you continue an entry from one line to the next, the second line of the entry must start with a tab character and a colon. The last line of a printcap entry must terminate with a colon.
- ▶ `\` (backslash) at the end of a line indicates that the entry continues on the next line. The backslash character must be followed, without blank space, by a carriage return.
- ▶ `lp=` specifies the file to be opened for output. Because you are printing a remote job, the `lp=` field for an AppleTalk printer printcap entry must be blank.
- ▶ `rm=` specifies the remote machine (above, `GatorPrinter`) to which the print job will be sent. The `rm=` field must specify the name of the GatorBox exactly as it appears in the `/etc/hosts` file so that the UNIX host can identify the IP address associated with the device. The

designated GatorBox must be configured for GatorPrint before it can function as a printer gateway.

- ▶ `rp=` specifies the logical name by which the AppleTalk printer will be identified from the UNIX side.
- ▶ `sd=` specifies the name of a spool directory. The spool directory identifies where the local `lpd` will store print jobs that are waiting to be printed. If you use this option in your `printcap` file, you should verify that the specified directory (for example, `/usr/spool/lpd/GatorPrinter/Franklin_PS`) exists with the proper permissions before you use the AppleTalk printer.

You should specify a separate spool directory for each logical printer in the `printcap` file. If two printers share the same spool directory, files intended for one printer may be output by the other printer. The `lpd` looks in its spool directory when it processes a job to see if there is anything else to print. If the `lpd` finds a file in the directory, it prints the file to the same printer to which it sent its last job. We recommend including the name of the GatorBox in the spool directory path name to ensure that print jobs routed through two GatorBoxes do not reference the same spool directory.

- ▶ `lf=` specifies where errors logged to standard error output should go. If you use this option in your `printcap` file, you should verify that the specified directory (for example, `/usr/adm/lpd-errors`) exists and can be written to by the line printer daemon.
- ▶ `mx=` specifies the maximum file size in blocks. If you specify `mx=0` (or if you omit the `mx` parameter), the file size is unlimited.



For a more detailed explanation of `printcap` entry formats, display the manual pages for `printcap` by typing `man printcap` on your UNIX system or refer to the Berkeley Systems Distribution documentation.

Examples

Unique logical printers for each host

Because the GatorBox can only accept one connection for each logical printer, you may want to create a separate logical printer for each host.

Given a site with three UNIX hosts, you might set up the following printcap entries:

```
#entry in /etc/printcap on HostA
Q1|q1:\
    :lp=:rm=GatorPrinter:sd=/usr/spool/lpd/q1:\
    :lf=/usr/adm/lpd-errs:\
    :rp=QUEUE1:

#entry in /etc/printcap on HostB
Q2|q2:\
    :lp=:rm=GatorPrinter:sd=/usr/spool/lpd/q2:\
    :lf=/usr/adm/lpd-errs:\
    :rp=QUEUE2:

#entry in /etc/printcap on HostC
Q3|q3:\
    :lp=:rm=GatorPrinter:sd=/usr/spool/lpd/q3:\
    :lf=/usr/adm/lpd-errs:\
    :rp=QUEUE3:
```

You would then create three logical printers (QUEUE1, QUEUE2, and QUEUE3) in GatorKeeper to reflect these /etc/printcap entries.

Printer LPR Name:	QUEUE1
LocalTalk Printer Name:	Franklin
LocalTalk Printer Type:	LaserWriter
LocalTalk Printer Zone:	*
Printer LPR Name:	QUEUE2
LocalTalk Printer Name:	Franklin
LocalTalk Printer Type:	LaserWriter
LocalTalk Printer Zone:	*
Printer LPR Name:	QUEUE3
LocalTalk Printer Name:	Franklin
LocalTalk Printer Type:	LaserWriter
LocalTalk Printer Zone:	*

One UNIX host sends print jobs to the GatorBox

As an alternative to setting up separate logical printers for each host, you can use one UNIX host ("Host A") to send print jobs to GatorPrint and set up your other UNIX hosts to send print jobs to Host A:

```
#entry in /etc/printcap on HostA
Q1|q1:\
    :lp=:rm=GatorPrinter:sd=/usr/spool/lpd/q1:\
    :lf=/usr/adm/lpd-errs:\
    :rp=QUEUE1:

#entry in /etc/printcap on HostB
Q2|q2:\
    :lp=:rm=HostA:sd=/usr/spool/lpd/q2:\
    :lf=/usr/adm/lpd-errs:\
    :rp=Q1:

#entry in /etc/printcap on HostC
Q3|q3:\
    :lp=:rm=HostA:sd=/usr/spool/lpd/q3:\
    :lf=/usr/adm/lpd-errs:\
    :rp=Q1:
```

You would then only need to create one logical printer (QUEUE1) in GatorKeeper:

Printer LPR Name:	QUEUE1
LocalTalk Printer Name:	Franklin
LocalTalk Printer Type:	LaserWriter
LocalTalk Printer Zone:	*

What NOT to do

To illustrate the wrong way to set up your print configuration, assume that you have the following /etc/printcap entries:

```
#entry in /etc/printcap on HostA
Q1|q1:\
    :lp=:rm=GatorPrinter:sd=/usr/spool/lpd/q1:\
    :lf=/usr/adm/lpd-errs:\
    :rp=QUEUE1:
```

```
#entry in /etc/printcap on HostA
Q2|q2:\
    :lp=:rm=GatorPrinter:sd=/usr/spool/lpd/q2:\
    :lf=/usr/adm/lpd-errs:\
    :rp=QUEUE1:
```

```
#entry in /etc/printcap on HostA
Q3|q3:\
    :lp=:rm=GatorPrinter:sd=/usr/spool/lpd/q3:\
    :lf=/usr/adm/lpd-errs:\
    :rp=QUEUE1:
```

and one logical printer (QUEUE1):

Printer LPR Name:	QUEUE1
LocalTalk Printer Name:	Franklin
LocalTalk Printer Type:	LaserWriter
LocalTalk Printer Zone:	*

With this printer configuration, your three UNIX hosts would contend for one print queue (QUEUE1). When more than three jobs were submitted, the lpd would send the excess jobs into a four-minute backoff, creating printer performance problems.



Chapter 7

NFS-to-AppleShare File Sharing

Network File System (NFS)

AppleShare

GatorShare



What is file sharing?

A *file server* is a device that lets network users ("clients") store and share files, folders, and applications. By opening a connection to a file server, a user can use and copy files and folders on a remote server volume. Storing files on a central file server lets network users exchange files with users in other locations easily. Because file servers typically have much larger storage capacity than client machines, using servers for file storage provides network users with more storage capacity than their own hard disks can offer.

The Network File System (NFS) is a network service developed by Sun Microsystems that lets users share file systems over a network. NFS uses a system of *mount points* to link remote directories to a user's local directory structure. Once linked, a remote directory can be accessed and updated as if it were connected directly to the user's local computer.

AppleShare is the network file service developed by Apple Computer for AppleTalk networks. The AppleShare File Server software lets a Macintosh with a hard disk function as the file server for client devices on an AppleTalk network. When a Macintosh is set up as an AppleShare file server, each hard disk corresponds to a volume on the server. Servers with more than one hard disk can present users with multiple volumes.

GatorShare gives Macintosh users access to NFS file service without taking away the familiar Macintosh interface. GatorShare lets you mount (link to) a remote mount point (directory) on an NFS file server and use it the way you would a local Macintosh disk. GatorShare translates the remote NFS server's directory structure into the Macintosh folder/file structure. Once mounted, the NFS mount point appears as an AppleShare volume on the Macintosh desktop. The mount point can be "opened" by double-clicking, and files and directories appear as file and folder icons in the mount point window. When a user double-clicks an icon representing an NFS directory, the GatorBox translates the "open folder" request to an NFS request, retrieves the list of files and folders within the NFS directory, and returns the directory contents to the Macintosh client in folder/file icon format.

A network user can open several volumes on the same server or volumes on several different servers at the same time. For example, Figure 7-1 illustrates how Rudolph can mount volumes from three servers (Dasher, Dancer, and Prancer) through the GatorBox.

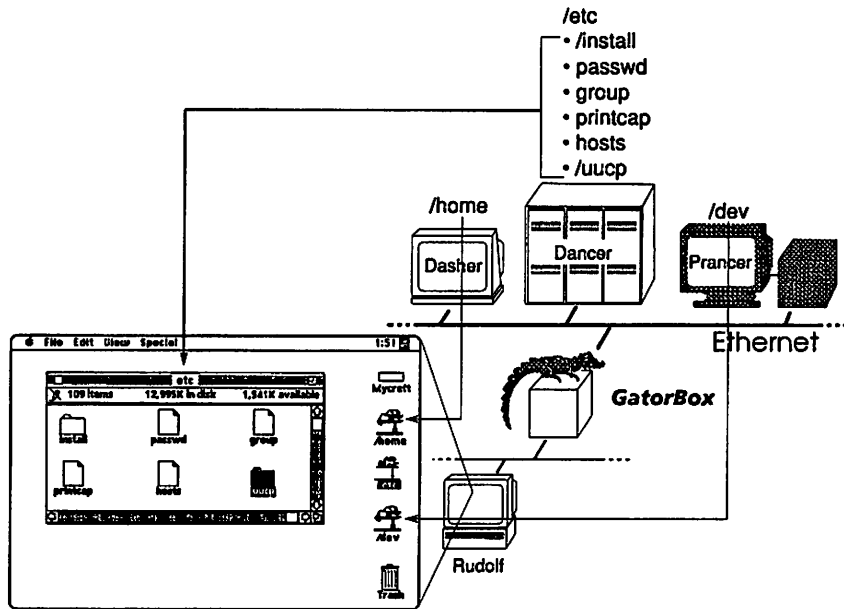


Figure 7-1. Mounting GatorShare volumes from several servers

About Network File System (NFS)

NFS (Network File System) was developed to simplify file sharing in networks that link dissimilar machines and operating systems. NFS lets users share files by mounting (linking to) file systems on other machines in the network. Once a file system is mounted, you can treat it as if it were a volume directly connected to your own computer. For example, you can create or read a file on the remote machine by using the same commands you would use on your local computer.

Directories and pathnames

UNIX file systems use hierarchical directories to store files. A directory is a file that contains the names and locations of files and other directories. At the top of the file hierarchy is the root directory, which is identified by a single slash (/) characters. Subdirectory names start with a slash (/)

character. One directory can be nested inside another directory to simplify organization of files. For example, Figure 7-2 shows a text file (*letter1*) inside a user's personal directory (*/johnson*), which in turn resides in the */home* directory on the NFS host.

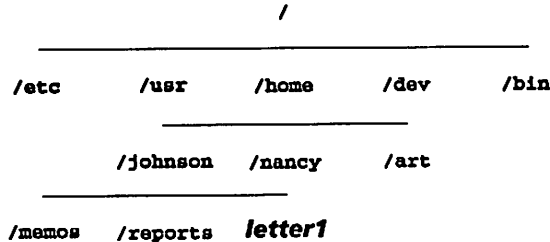


Figure 7-2. Sample NFS directory structure

When files are stored in directories, they are identified by their *pathnames*. Pathnames describe how to reach files on the file system. For example, the pathname for the *letter1* file in Figure 7-2 would be */home/johnson/letter1*.

Mount points

NFS clients access files on a remote server by mounting the directories in which those files reside. An NFS client mounts a remote directory by linking it to a *mount point* in its local directory structure. A mount point is a location (typically a directory) in the client's file system to which the client attaches the remote directory. Once attached, the mount point temporarily "owns" the mounted directory, and files on the remote server appear to reside within the mount point.

Export lists

Before an NFS client can mount a directory on an NFS server, the server must *export* the available directories to the client. Exporting means that an NFS server advertises the list of available directories to its clients. The network administrator controls which directories on the NFS server can be mounted by editing the */etc/exports* file.

/etc/exports file

The `/etc/exports` file lists the directories on an NFS host that can be exported to NFS clients. You cannot include different levels of a directory structure in the `/etc/exports` file. For example, you cannot include both `/etc` and `/etc/usr` in the `/etc/exports` file.

The `/etc/exports` file can restrict the type of access or the clients that are authorized to access an exported directory. For example, the `-ro` argument in the first line in Figure 7-3 specifies that an NFS client that mounts the `/` (root) directory can exercise read-only access. The second line specifies that host `vixen` has read/write access to the `/etc` directory; this implies that other hosts have read-only access. If you do not specify an access restriction, the system provides read/write access to authorized users.

You can restrict access to a directory to specified hosts by entering host names on the same line as the directory. For example, the third line in Figure 7-3 specifies that only users on host `dasher` can access files in the `/usr` directory. If you do not specify access restrictions, the system lets anyone on the network mount the directory.

```
/          -ro
/etc       -rw=vixen
/usr       -access=dasher
```

Figure 7-3. Sample `/etc/exports` file entries



If you are logged into an NFS server, issue an `exportfs` command to list the file directories the server exports. If you are not logged into the NFS server, issue a `showmount -e <servername>` command to display the server's export list.

Required NFS daemons

NFS requires that the `portmapper`, `nfsd`, and `mountd` daemons be running on any server on which one or more mount points will be accessed.

portmapper

The `portmapper` daemon tracks the relationship between ports and services on an NFS host. When a network service is assigned a port number, it registers its port number with `portmapper`. When a client wants to use the network service, it obtains the service's port number from `portmapper`. Once it has the correct port number, the client can call the server.

nfsd

The `nfsd` daemon runs on an NFS server and handles file system mount requests from NFS clients.

mountd

The `mountd` server daemon is the NFS mount request server. `mountd` reads the `/etc/exports` file when it receives a request for a remote mount. The `/etc/exports` file contains the entries for directories that are currently exported.



Issue a `ps -ax` or `rpcinfo -p` command on your BSD UNIX system to display the daemons running on a server. Issue a `ps -ef` command on your System V UNIX system to display running daemons.

NFS security

NFS security places limits on the clients that can mount directories on an NFS server, the users on a client who can access files, and the types of access authorized users can exercise. In network environments that do not use the Network Information System (NIS) to propagate authentication information, an NFS server uses the `/etc/hosts`, `/etc/passwd`, and `/etc/group` files to validate access. In most situations, a host will use the files in its own directory structure. However, one NFS host can authenticate user access by comparing a user's login name and password to the `/etc/passwd` and `/etc/group` files on a second NFS host.

/etc/hosts file

The `/etc/hosts` file holds the names and IP addresses of hosts on a network. Programs use the `/etc/hosts` file to locate other machines when they need to communicate over a network. If a remote machine wants to access a file on an NFS host, the remote machine's name and IP address must be entered in the NFS host's `/etc/hosts` file.

Figure 7-4 presents a sample `/etc/hosts` file entry.

<u>192.31.224.12</u>	<u>frodo Frodo timehost</u>	<u>#Sun 3/280 w SunOS4.0</u>
IP address	Host names	Comment

Figure 7-4. Sample `/etc/hosts` file

You must enter the GatorBox's name and IP address in the server's `/etc/hosts` file before GatorShare can mount volumes on the server. Because `/etc/host` entries are case sensitive, the name should match exactly the name as it appears in the GatorBoxes window in GatorKeeper.

User access security

After the system verifies that an NFS connection between an NFS server and a remote client is authorized, the system tests whether a user can access information on the server. User access security involves a comparison of the login name and password entered by the user with the login names and passwords set up in the server's `/etc/passwd` file. If the user enters a valid name and password, he or she receives the file access appropriate to that login name and the group associated with the login name.

`/etc/passwd` file

The `/etc/passwd` file lists the login name, password, and home directory for every user authorized to access an NFS host. Figure 7-3 illustrates a sample `/etc/passwd` file entry.

<u>batguy</u>	<u>:iH9MnOfq4LsQQ</u>	<u>:33</u>	<u>:23</u>	<u>Bruce Wayne</u>	<u>:/home/batguy</u>	<u>:/bin/csh</u>
login name	User password (encrypted)	User ID (UID)	Group ID (GID)	Comment	Default home directory	login shell

Figure 7-5. Sample `/etc/passwd` file entry

In a network environment that uses NIS (Yellow Pages) to distribute user authentication information, the following entry is added to the `/etc/passwd` file:

`+:0:0:0::`

When a user logs in with an identifier not found in the local host's `/etc/passwd` file, the host queries NIS to validate the user's password.

Some operating systems, such as AIX, now offer shadow passwords for more secure user authentication. When shadow passwording is used, user names and passwords are stored in a separate file. The `/etc/passwd` file entry for a user has `##<login name>` in place of the user's encrypted password.

`/etc/group` file

User groups let you set up file access for sets of users, such as members of a department or work group. The `/etc/group` file lists the names and ID numbers of network user groups and the names of users that belong to each group. Figure 7-4 presents a sample `/etc/group` file entry.

```
support:*:23:gordon,michael,chris,larry,wang,carlos,throop
```

The diagram shows the entry `support:*:23:gordon,michael,chris,larry,wang,carlos,throop` with three labels and brackets below it:
 - `support` is labeled "Group name"
 - `:23:` is labeled "Group ID (GID)"
 - `gordon,michael,chris,larry,wang,carlos,throop` is labeled "login names"

Figure 7-6. Sample `/etc/group` file

In a network environment that uses NIS to distribute user authentication information, a plus (+) sign is added as an entry to the `/etc/passwd` file. When a user logs in with an identifier not found in the local host's `/etc/group` file, the host queries NIS to validate the user's group identifier.

File access security

After a host has verified that a user is authorized to log in, the user can display and modify files and run programs on the host. The files and programs to which the user has access are controlled by the host's file access security. File access security involves a comparison of the user's ID and group with the user ID and group ID associated with each file. NFS recognizes three classes of users for a file:

- ▶ **Owner** — Typically the person who created the file or directory
- ▶ **Group** — Typically the group of users to which the owner belongs

- **All** — Any user except the owner and members of his or her group

When a user creates a file or directory, he or she can set access restrictions to specify whether each class can display (read), modify (write), or run (execute) the file. For example, assume that Maria, a member of the Marketing group, creates a memo called `new_ad`. She wants other members of her group to be able to read the memo, but she doesn't want them to change it. Because the memo contains confidential information, she doesn't want users outside her group to be able to read it.

To limit access to the new file, Maria assigns it a permission code of `rwX r-- ---`.

- The first three characters (`rwX`) specify that the owner (Maria) can read, write, and execute the file.
- The second three characters (`r--`) specify that the file's group (Marketing) can read (but not change) the file.
- The third three characters (`---`) specify that users other than Maria or her group cannot read the file.

File access permissions are modified with the `chmod` command.

Network Information System (NIS)

The Network Information System is a read-only database look-up service. NIS administers host, password, and group information for machines on a network. Use of NIS simplifies system administration, since an administrator can set up user authentication in one central location and propagate it throughout the network instead maintaining host and user information on each machine on the network.

NIS uses the following files to authenticate user access to hosts on the network:

- The `hosts` database consists of the names and IP addresses of machines on the network.
- The `passwd` database, like the `/etc/passwd` file, consists of the usernames, passwords, and related information about users.

- ▶ The group database, like the `/etc/group` file, consists of the names and identification codes of user groups.
- ▶ The aliases database lists the alternate names for users on the network.



The Network Information System used to be known as Yellow Pages (or YP). In 1989, Sun was advised by British Telecom that "Yellow Pages" was a protected trademark. Sun renamed its Yellow Pages service to Network Information Service. NIS and Yellow Pages are functionally identical — although the term "Yellow Pages" is no longer used, many of the files and commands it uses still begin with "YP."

How NIS works

When a user enters a password, the local host issues a `ypmatch` command to the NIS server for its domain. The `ypmatch` command specifies the user's login name and encrypted password. The NIS server compares the encrypted password to the appropriate entry in its `passwd` file. If the encrypted password submitted with the `ypmatch` command corresponds to the encrypted password in the `passwd` file, the NIS server returns the user ID and group ID associated with the login name.



Use the `ypwhich` command to identify the NIS server for your local machine. Use the `ypwhich <hostname>` command to display the name of the NIS domain for another host on your network.

PCNFSD

PCNFSD is a UNIX daemon that lets a central host validate a user's login name and password without implementation of NIS. Unlike NIS, PCNFSD can run on any compatible host on the network. Most NFS implementations, such as Sun workstations, support PCNFSD authentication.

When you specify that a GatorBox should use PCNFSD authorization, GatorShare passes the user's name and password to the PCNFSD daemon. The daemon performs the authentication and returns the `uid` (user ID) and `gid` (group ID) of the user. Unlike NIS, PCNFSD does not return the user's home directory. Consequently, use of PCNFSD authorization may require that a site set up mount points for each directory that holds a user's home directory.

PCNFSD is most useful to sites that implement shadow passwords or password aging. Under password aging, the password for a user's entry in the `/etc/passwd` file is followed by a comma and an expiration date. GatorShare submits the entire `<encrypted password>`, `<expiration date>` string to the PCNFSD daemon, which

PCNFSD is also necessary for sites that don't use the standard UNIX `passwd` and `group` files or that have experienced extremely long authentication times.

Using PCNFSD authentication has some limitations:

- ▶ You will no longer see the User's Home Directory as an option in the Chooser's volume list when you mount a volume. This is because this information is not passed back by the PCNFS daemon on the NFS host and therefore can't be ascertained by the GatorBox.
- ▶ The Get Privileges function in the Finder will show the owner and group name of a folder by User ID (`uid`) and Group ID (`gid`), not by the actual names. Consequently, in order to change these, the `uid` and `gid` of the new owner or group must be known.

AppleShare

Macintoshes using the AppleShare workstation (client) software use and store files on Macintoshes set up as AppleShare servers. Macintosh users organize files on their disks by putting them into *folders*. A folder can contain other folders, letting a user "nest" folders to organize information hierarchically. For example, a Macintosh user can set up a folder called "Correspondence," which contains separate subfolders for letters written in January, February, and so on. The folder for each month might contain other folders ("Status memos", "Letters", "Personal") or files ("Letter to Dave", "Jan17 Status").

File formats

Macintosh files consist of two forks:

- ▶ The **resource fork** holds information used by the Macintosh operating system, such as the appearance or placement of icons, menus, or dialog boxes.

- The **data fork** holds the actual data for the file, such as the text in a letter or the numbers in a spreadsheet.

Apple has defined two formats for storing Macintosh files on foreign (non-AppleShare) file systems:

- **AppleSingle** — Macintosh files in AppleSingle format are stored with data and resource information in separate parts of the same file. Because non-Macintosh applications have difficulty reading AppleSingle files, and because AppleSingle files cannot be easily modified, an AppleSingle file format should only be used to store archival backups.



AppleSingle should be used only for archive purposes. You can copy Macintosh files to or from an AppleSingle volume at any time. However, you should not create a file on an AppleSingle volume or use a Macintosh application and GatorShare to modify a file residing on an AppleSingle volume.

- **AppleDouble** — Macintosh files in AppleDouble format are stored with data and resource information in separate files. Non-Macintosh applications ignore the resource file and read only the data file.

PC AppleShare

IBM-compatible personal computers (PCs) equipped with Apple's PC AppleShare client software and the appropriate network interface card can access folders and files on the NFS server in the same way that Macintosh users do. Where Macintosh users see a series of nested folders and files, PC users see directories, subdirectories, and files.

Because of a bug in version 2.02 of Apple's PC AppleShare client software, DIR listings of the top level of a mounted volume display only the creation dates and times for files. The same files viewed from a Macintosh will show the modification dates and times. This tends to make incremental backup of these files difficult. At lower levels of the mounted volumes, both the PC and the Macintosh will see the modification dates and times.

PCs running PC AppleShare will be unable to see files that have the "Bundle", "System", and "Inited" bits turned on, such as the Users&Groups file on an AppleShare server or the Easy Access file.

AppleShare security

Since many users on the network may be storing files on the server, AppleShare needs a way of restricting access to files on the server to authorized users. As part of logging on to an AppleShare server, a user enters his or her name and password. The AppleShare Server software determines whether the user is authorized to log in by comparing the user's name and password to entries in a database maintained by the system administrator. If a match is found, AppleShare lets the user access volumes on the specified server.

AppleShare access privileges

The permissions and access restrictions in AppleShare apply only to folders. All files inherit the permissions of the folder in which they reside. On an AppleShare server, each folder is associated with an *owner* and a *group*:

- ▶ A folder's owner (that is, the person who created the folder or who was assigned ownership) can specify whether other users can read or modify files inside the folder.
- ▶ A folder's group is a set of users set up by the system administrator who have special access privileges to files in the folder. For example, a folder might be set up so that its contents can be read only by members of the Marketing group.

Not only can you control *who* has access to folders, but you can also control *how* users access them. In AppleShare, users can have the three types of access permission:

- ▶ **See Folders** lets a user see the icons of subfolders within the protected folder; whether the user can see the contents of the subfolder depends on the permissions associated with the subfolder.
- ▶ **See Files** lets a user see the icons of documents and applications within the protected folder and read or copy the contents of those files.
- ▶ **Make Changes** lets a user modify the files within the protected folder.

Folder permissions can be set for the folder owner, the folder group, or everyone. Unlike NFS, however, AppleShare does not allow access

restrictions on individual files. A file inherits the access restrictions of the folder (directory) in which it resides.

For example, if Maria wants members of the Marketing group to be able to see the files in a folder containing marketing brochures but not make changes, she would set the permissions on her folder to *See Files*, *See Folders*, *Make Changes* for the owner (herself); *See Files* and *See Folders* for the Marketing group; and no permissions for everyone outside the Marketing group. If someone outside the Marketing group mounted the volume on which Maria's folder resides, the icon of the folder will be dimmed (grayed). Double-clicking the icon will result in an access violation error.

GatorShare

GatorShare makes specified directories on NFS file servers look like AppleShare server volumes to Macintoshes on the AppleTalk network. Client Macintoshes send a standard AppleShare request, such as a request for a list of the files and folders on a directory, to the GatorBox. The GatorBox translates this AppleShare request into the appropriate NFS request. When the NFS server sends a reply to a user's request, GatorShare translates the NFS reply into a format that the Macintosh can understand. The Macintosh client then displays all the files and directories like standard Macintosh files and folders.

Macintosh users interact with the NFS server as if it was an AppleShare server without realizing that the GatorBox is "fronting" for the NFS server. When a Macintosh user mounts an NFS volume, GatorShare establishes a connection to the NFS server and issues an NFS mount request. When the `nfsd` server process receives the mount request, it checks the information in the request for consistency and security authentication. If the `nfsd` process validates the request, the NFS server informs GatorShare that the user is authorized for access and provides a description of the mount points available on the host.

The Macintosh user perform standard operations such as copying files or launching applications. For example, when a user double-clicks a folder icon in a mounted AppleShare volume, the user's Macintosh sends a request to the AppleShare server for information about that folder. When the information is returned, the Macintosh displays the files and folders within that first folder. When a user mounts a GatorShare "volume," the user

actually mounts a directory on an NFS server. A folder icon on the GatorShare volume represents a subdirectory of the mounted directory from the NFS server.

.DESKTOP file

The Macintosh Finder uses an invisible file called Desktop to map files to their applications and icons. Whenever files are written to a Macintosh volume, the Finder updates the Desktop file with the new file information.

GatorShare creates a .DESKTOP file for each NFS mount point. The GatorShare .DESKTOP file is a database that stores information, such as the location of folders and files, required to display the Macintosh desktop. The .DESKTOP file can be stored on the mount point's server or on another machine on your network. You can store the .DESKTOP file in a directory below the mount point root or in a directory on a different mount point on the same server.

File creation times/dates

A Macintosh file or directory stores its creation time/date, modification time/date, and backup time/date as part of its file header information. When a user creates or modifies a file on an NFS server through GatorShare, the GatorBox uses the UDP time service to determine the correct date and time for these fields. If the UDP time service is not available, the GatorBox "touches" the .DESKTOP file to determine the current time and date.



To verify that the time service on a host is active, confirm that the `/etc/inetd.conf` file includes an enabled time entry. If it does, issue a `ps -ax` command to verify that `inetd` is running.

File name mapping

Because the Macintosh operating system accept more characters in file names than most other systems, GatorShare lets you specify how characters in Macintosh file names should be converted, or mapped, when the file is copied to the NFS file system. GatorShare replaces each illegal character in a Macintosh filename with a three-character string: a *delimiter character* specified by the network administrator for the volume a two-character hexadecimal code for the illegal character.

GatorShare provides four options for file name mapping:

- ▶ **No filename mapping** — If you specify *No filename mapping*, GatorShare will not convert characters in filenames when Macintosh files are moved to an NFS server. If a filename includes one or more unacceptable characters, GatorShare will generate a diagnostic message but will not copy the file.
- ▶ **8-bit filenames** — If you specify *8-bit filenames*, GatorShare will allow all 8-bit ASCII characters except slash (0x2F), null (0x00), or the delimiter character in filenames when Macintosh files are moved to an NFS server. For example, if the delimiter character is a colon, GatorShare would convert the Macintosh file name GatorBox CS/Rack™ to GatorBox CS:2FRack™.
- ▶ **7-bit filenames** — If you specify *7-bit filenames*, GatorShare will allow all 7-bit ASCII characters except a slash (0x2F), null (0x00), or delimiter character in filenames when Macintosh files are moved to an NFS server. For example, if the delimiter character is a colon, GatorShare would convert the Macintosh file name GatorBox CS/Rack™ to GatorBox CS:2FRack:AA.
- ▶ **7-bit alphanumeric filenames** — If you specify *7-bit alphanumeric filenames*, GatorShare will allow only alphanumeric (0-9, a-z, A-Z) characters, the underscore character, and the last period in a file name when Macintosh files are moved to an NFS server. For example, if the delimiter character is a colon, GatorShare would convert the Macintosh file name GatorBox CS/Rack™ to GatorBox:20CS:2FRack:AA.

Figure 7-7 lists the hexadecimal equivalents for 7- and 8-bit characters.

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
7-bit characters	00																
	10																
	20		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
	30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
	60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8-bit characters	80	Ä	Å	Ç	É	Ñ	Ö	Ü	á	à	â	ä	ã	ä	ç	é	è
	90	ê	ë	í	ì	ï	ñ	ó	ò	ô	ö	õ	ú	ù	û	ü	
	A0	†	°	¢	£	§	•	¶	ß	®	©	™	'	"	≠	Æ	Ø
	B0	∞	±	≤	≥	¥	μ	∂	Σ	Π	π	∫	•	•	Ω	æ	ø
	C0	¿	¡	¬	√	ƒ	≈	Δ	«	»	...	À	Ã	Ö	Œ	œ	
	D0	-	-	"	"	'	'	+	ó	ÿ	ÿ	/	□	<	>	fi	fl
	E0	‡	•	,	"	%o	Â	Ê	Á	Ë	È	Í	Î	Ï	Ì	Ó	Ô
	F0		Ò	Ú	Û	Ü	ı	ˆ	˜	-	˘	˙	˚	¸	˝	˞	˜

Figure 7-7. Character mapping table

Byte-range locking

Byte-range locking allows multiple users to open and update a file residing on a remote server simultaneously. This affects sites using Macintosh applications such as Microsoft Excel 2.2 or Informix' Wingz, which commonly have several people using the same files at the same time.

Byte-range locking enables use of multi-user applications on an individual GatorBox basis. Multiple users who wish access to the same file must mount the volume from the same GatorBox (that is, the users must select the same zone in the Chooser and the same server from the Server list, and they must pick the same volume from the list of mount points.)

GatorShare security

Before a user can access a file server through the GatorBox, GatorShare verifies that the user has the appropriate permissions. A Macintosh user accesses an NFS file server by clicking the AppleShare icon in the Chooser. When the Chooser displays the list of file servers available in the specified zone, the user selects one or more server names and enters his or her name and password.

User access matching

When a user enters a password in the AppleShare Access dialog box, the GatorBox encrypts the password. The GatorBox issues a remote procedure call (RPC) to the target host to identify the NIS server (if any). When GatorShare reaches the appropriate NIS server, it issues a second RPC that asks the server to return the user's `/etc/passwd` entry. GatorShare then compares the password portion of the `/etc/passwd` entry to the encrypted version of the password the user entered. If the two passwords match, the user is authorized to access the host. The AppleShare client software displays a list of volumes available on the specified server, and the user selects one or more volumes to mount.



Chapter 8

GatorBox Administration

TELNET shell

Simple Network Management Protocol (SNMP)

C

C

C

TELNET shell

You can TELNET to the GatorBox to check whether it is running and to query it about many of its services. You initiate a TELNET connection by issuing a `telnet <GatorBoxIP Address>` command. If the GatorBox has been assigned a password, you will be prompted to enter the password before you can complete the TELNET connection to the GatorBox.

TELNET syntax

The TELNET syntax for querying a GatorBox is:

```
help      quit
          show
          status

reset     alap
          enet

restart

show      ip [routes|arp]
          appletalk [routes|arp|zones|interfaces]
          decnet [nodes|circuits|status]
          share
          alap
          enet
          log
          crash
          memory
          dump

status

quit
```

Sample TELNET commands

reset alap

reset alap clears the alap statictics in the GatorBox.

reset enet

reset enet clears the Ethernet statictics in the GatorBox.

restart

restart causes the GatorBox to restart.

show ip routes

show ip routes displays the IP routes stored in the GatorBox.

```
Dogzilla> sho ip routes
```

```
IP gateway (route) table:
```

0. Dest 192.31.157.0, gateway 192.31.50.144, cost 1, timeout 1, (via RIP)
1. Dest 192.31.240.0, gateway 192.31.50.90, cost 1, timeout 1, (via RIP)
2. Dest 192.31.53.0, gateway 192.31.50.195, cost 1, timeout 1, (via RIP)
3. Dest 192.31.127.0, gateway 192.31.50.109, cost 1, timeout 1, (via RIP)
4. Dest 192.31.170.0, gateway 192.31.50.49, cost 1, timeout 1, (via RIP)
5. Dest 192.31.154.0, gateway 192.31.50.53, cost 1, timeout 0, (via RIP)

```
IP route cache:
```

- 0: Net 192.31.1.128, gateway 192.31.50.18, timeout 10
- 17: Net 192.31.1.113, gateway 192.31.50.18, timeout 10
- 19: Net 192.31.1.211, gateway 192.31.50.18, timeout 10
- 21: Net 192.31.1.21, gateway 192.31.50.18, timeout 5
- 25: Net 192.31.50.249, gateway 192.31.50.249, timeout 10
- 25: Net 192.31.1.121, gateway 192.31.50.18, timeout 10
- 31: Net 192.31.50.255, gateway 0.0.0.0, timeout 9

show ip arp

`show ip arp` displays the Ethernet address resolution table stored in the GatorBox.

```
Dogzilla> sho IP arp
Ethernet IP ARP table:
0: IP 192.31.50.126 Hardware 8.0.20.a.aa.88 (flags 0x1)
2: IP 192.31.50.110 Hardware aa.0.4.0.2.4 (flags 0x1)
3: IP 192.31.50.249 Hardware aa.0.4.0.2.4 (flags 0x1)

MacIP Address assignment table:
0: IP 192.31.54.2, net 50, node 76
1: IP 192.31.54.3, net 50, node 107
3: IP 192.31.54.5, net 50, node 38
```

show appletalk routes

`show appletalk routes` displays the AppleTalk routes stored in the GatorBox.

```
Dogzilla> sho appletalk routes

AppleTalk routing table:
#0: net 1239-1239, dist 2, port 2, bridge ip: 192.31.1.113
    port 910, flags 0x400, type 0x2
    zone 'Front Desk', use 1
#1: net 1382-1382, dist 1, port 1, bridge atalk:
    (22200.252.1), flags 0x400, type 0x0
    zone 'Support:Leslie', use 1
#2: net 666-666, dist 2, port 1, bridge atalk:
    (22200.72.1), flags 0x400, type 0x0
    zone 'Hyphenation', use 1
#3: net 22000-22999, dist 0, port 1, bridge unset, flags
    0x500, type 0x0
    zone 'Caribbean', use 2
#4: net 62249-62249, dist 1, port 2, bridge ip:
    192.31.1.121 port 910, flags 0x400, type 0x2
    zone 'Gumby', use 1
```


show appletalk arp

`show appletalk arp` displays the AppleTalk address resolution information stored in the GatorBox.

```
Dogzilla> show appletalk arp
AppleTalk AARP cache
#   flags age pending  net node len addr
```

show appletalk zones

`show appletalk zones` displays the AppleTalk zone table stored in the GatorBox.

```
Dogzilla> sho appletalk zones
AppleTalk zone list:
zone 'Treadmill', use 1
zone 'walla walla', use 1
zone 'Real Work', use 1
zone 'Combat', use 1
zone 'Talk, talk', use 1
zone 'Isolation', use 1
zone 'Slalom Zone', use 1
zone 'Postal', use 1
zone 'Engineering', use 1
zone 'Caribbean', use 2
zone 'Ethertalk', use 2
zone 'Hyphenation', use 2
```

show appletalk interfaces

`show appletalk interfaces` displays the AppleTalk interface information for the GatorBox.

```
Dogzilla> sho appletalk interfaces
AppleTalk Router flags: none (0x0)
AppleTalk Router state: up
```

```
AppleTalk Ports:
LAP 0:
  State (initialized)
  Status (Phase-2 Enabled Node-valid Network-valid
    Zones-valid)
```

```
Address (50.128), Network 50, Node hint 128
zone 'School', use 2
```

```
80220:
State (initialized)
Status (Phase-2 Long-DDP-headers Extended-Net
Use-Multicast Enabled Node-valid
Network-valid Zones-valid)
Address (22200.68), Network 22000-22999, Node hint
(22200.68), Default zone 'Caribbean'
zone 'Caribbean', use 2
```

```
IP 1:
State (initialized)
Status (Encapsulated Half-bridge Long-DDP-headers
Enabled Node-valid Network-valid Zones-valid)
Address (0.0), Network 0, Node hint 0
```

show decnet nodes

show decnet nodes displays information about the DECnet nodes known to the GatorBox.

```
Dogzilla> sho decnet nodes
DECNet level 1 router; Address 37.100, Priority 1

Known nodes:
node 37.111, ethernet end node hops 1, cost 4 next node
37.111
```

show decnet circuits

show decnet circuits displays information about the DECnet circuits known to the GatorBox..

```
Dogzilla> sho decnet circuits
Known Circuits:
DECNet level 1 router; Address 37.100, Priority 1
Port 0, driver LAP 0
Link cost 8
Maximum segment size (MSS) 542
Circuit statistics:
Transit received 0
Transit sent 0
```

```
Total received 0
Local received 36
Originating sent 0
Transit congestion 0
Terminating congestion 0
Circuit down 0
Init failure 0
```

```
DECNet level 1 router; Address 37.100, Priority 1
Port 1, driver ENET0
Link cost 4
Maximum segment size (MSS) 542
Circuit statistics:
Transit received 0
Transit sent 0
Total received 193047
Local received 193047
Originating sent 0
Transit congestion 0
Terminating congestion 0
Circuit down 0
Init failure 0
```

show decnet status

show decnet status displays the information about the status of the DECnet network.

```
Dogzilla> show decnet status
DECNet level 1 router; Address 37.100, Priority 1
Router statistics:
Node unreachable 0
Aged packets 0
Node out of range 0
Oversized packets 0
Packet format errors 0
Partial routing updates 0
Verification rejects 0

Area-wide maximum segment size 542
Area-wide maximum hop count 2
```

show share

`show share` displays a list of GatorShare users and the volumes they have mounted.

```
Dogzilla> show share
GatorShare Users...
# st refN User name
```

show alap

`show alap` displays the ALAP (AppleTalk Link Access Protocol) statistics for the GatorBox.

```
Dogzilla> show alap
ALAP driver statistics:
```

```
XmitCount 49450
UndCount 0
LineBusys 587
CollsnCount 713
DeferCount 53
IdleTOCount 0
EnableAbort 586
ExtIntCount 587
AbortIntCount 586
TopOfDefer 586
```

```
IntCount 10515
NoBuffCount 0
RcvCount 10463
CRCCount 0
OvrCount 50
```

```
LenErrCnt 0
BadCount 0
BadDDP 0
NoDtaCount 2
RandomCTS 0
```

show enet

show enet displays the Ethernet statistics for the GatorBox.

```
Dogzilla> show enet
Ethernet driver statistics:

Packets out:  62415
Packets in:   678037

Xmit errors:  0
Recv errors:  0

Buffer full:  10
No handler:   77
No Message:   14
Out of window collisions:  95
Aborted transmissions:    0
Transmission timeouts:   0
FIFO underruns:  0
```

show log

show log displays the next 25 lines of the GatorBox diagnostics log.

```
Dogzilla> show log
Message Log:
  Message Log:
5924310 L4 0000 AR: Timeout waiting for ARP response from
(22325.126)
5924310 L4 ffff019 AR: timeout waiting for ARP response
5928405 L4 0000 AR: too many pkts waiting for ARP response
from (22200.225)
5928405 L4 ffff01e AR: packet dropped; too many pkts
waiting for ARP
5928405 L4 0000 AR: too many pkts waiting for ARP response
from (22200.225)
5928405 L4 ffff01e AR: packet dropped; too many pkts
waiting for ARP
5928405 L4 0000 AR: too many pkts waiting for ARP response
from (22200.225)
5928405 L4 ffff01e AR: packet dropped; too many pkts
waiting for ARP
5928405 L4 0000 AR: too many pkts waiting for ARP response
from (22200.225)
```

show crash

`show crash` displays information about the last GatorBox crash.

```
Dogzilla> show crash
Crash dump:
Crash Signature 'vvvv',
Last crash PC ffffffff, SR ffff
stack:
ffff ffff
ffff ffff
ffff ffff
ffff ffff
ffff ffff
ffff ffff
ffff ffff
frame Pointers:
ffffffff
ffffffff
ffffffff
ffffffff
ffffffff
Registers:
A0 ffffffff      D0 ffffffff
A1 ffffffff      D1 ffffffff
A2 ffffffff      D2 ffffffff
A3 ffffffff      D3 ffffffff
A4 ffffffff      D4 ffffffff
A5 ffffffff      D5 ffffffff
A6 ffffffff      D6 ffffffff
A7 ffffffff      D7 ffffffff
```

show memory

`show memory` displays memory usage information for the GatorBox.

```
Dogzilla> show memory
GatorBox Memory Info...
Total bytes 2514100, free 2265202, allocated 248898
Text 491016, Data 12736, BSS 28552, Pool 41800
```

sho dump

`show dump` displays the information currently in the GatorBox's memory.

status

`status` displays the current status of the GatorBox.

```
Dogzilla> show status
GatorBox Status:

GatorBox CS TELNET shell v0.3
GatorShare version 2.0 (build 2)
Hardware Type: Cayman Systems Inc. GatorBox CS
Serial Number: 123456
Ethernet Address: 00.00.89.01.e2.40

Error Logger:
Low 2, Med 93, High 124, Warn 619
Lost msgs 0, total msgs 838
Boot state: running
```

Simple Network Management Protocol (SNMP)

The Simple Network Management Protocol (SNMP) is an asynchronous network management protocol. SNMP lets a network administrator monitor and correct problems on a TCP/IP network fetching and modifying settings on remote network devices. The network administrator typically runs an SNMP client program on a local host to obtain information from or send information to an SNMP server. For example, a network administrator can use SNMP to query a GatorBox about its ALAP statistics, AppleTalk or Ethernet routing table information, and AppleTalk zone information.

SNMP messages either specify that the server should retrieve information from its set of management variables and report the results to the client or stores values in the management variables. When a request is received from an SNMP client, the server locates the internal variables that correspond to the SNMP objects specified in the request. The server translates the SNMP request and performs the appropriate operations on its data structures.

SNMP uses four operators to communicate information between clients and servers:

- ▶ `get` is used to retrieve specific management information
- ▶ `get_next` is used to retrieve management information by means of array traversal
- ▶ `set` is used to manipulate management information
- ▶ `trap` is used to report unusual network events

SNMP uses a store-and-retrieve model to communicate information about a network. An SNMP server, such as a GatorBox, constantly updates internal arrays with information about network status and traffic. Some of the variables, such as the number of packets received, are simple counters. Other variables, such as the server's routing tables or ARP cache, are stored in more complex tables.



For more information on SNMP and related topics, refer to RFC 1065 ("SMI (Structure of Management Information)"), RFC 1066 ("MIB (Management Information Base)"), and RFC 1067 ("SNMP (Simple Network Management Protocol)").

Internet-standard MIB

SNMP uses an Internet-standard Management Information Base (MIB) to define the variables that SNMP servers are expected to support and the format for requests for each variable. Each MIB variable records one item of data, such as the status of a connected network, traffic statistics, counts of errors encountered, and IP routing tables.

Internet-standard MIB provides eight groups of management objects:

- ▶ The `system` group contains generic information about the configuration of the SNMP server, such as the device name and how long ago the server started. Variables in the system group start with a `sys` prefix.
- ▶ The `interfaces` group contains generic information on the interfaces attached to a server, such as the number of interfaces attached to an SNMP server and a description of each interface. Variables in the interfaces group start with an `if` prefix.

- ▶ The address translation group contains a table of address resolution information, such as the physical address and the IP address for a host. Variables in the interfaces group start with an `at` prefix.
- ▶ The IP group contains information about the datagrams received, forwarded, and discarded and the IP routes and addresses associated with a node. Variables in the IP group start with an `ip` prefix.
- ▶ The ICMP group consists of a series of counters, which record the number and type of each ICMP message sent and received by the local IP entity. Variables in the ICMP group start with an `icmp` prefix.
- ▶ The TCP group contains information about the TCP connections opened and attempted. Variables in the system group start with a `tcp` prefix.
- ▶ The UDP group contains information about UDP datagrams sent, received, and discarded. Variables in the UDP group start with a `udp` prefix.

Cayman's private MIB

The Internet-standard MIB represents the basic set of SNMP management objects. Cayman developed an extension to the standard MIB to encompass objects specific to the GatorBox. The Cayman private MIB provides 12 groups of management objects, including ALAP, RTMP, and NBP information. The GatorBox MIB, which is available via anonymous ftp from `ftp.cayman.com`, is listed in Appendix B of this manual.

Appendix A

Glossary

AARP

AppleTalk Address Resolution Protocol. *See* ARP.

access privileges

The permissions assigned to an AppleShare user that determine whether the user can read and/or modify the contents of files on a server.

active window

Frontmost window on the Macintosh screen; identified by highlighted title bar.

addressing

Method used by a network protocol to identify the source and destination nodes for data transmitted on a network.

address resolution

Conversion of an internet address into a corresponding physical address. *See* ARP.

adjacent

Node on the same physical network as the local node (DECnet).

ADSP

AppleTalk Data Stream Protocol.

AFP

AppleTalk Filing Protocol; responsible for access to files stored on a remote file server.

ALAP

AppleTalk Link Access Protocol; responsible for node-to-node delivery of data on a *single* AppleTalk network and for assigning unique node identifiers to each station on an AppleTalk network.

ALAP frame

Variable-length packet of data preceded and followed by control information. The *ALAP frame header* specifies the node identifiers of the frame's destination and source nodes, which are used to deliver and acknowledge transmission of frames. The *ALAP frame trailer* contains packet validation information, which is used by the receiving node to detect transmission errors.

angle brackets

Term for the < and > characters.

AppleDouble

File format that stores a Macintosh file's data and resource information in separate files. Non-Macintosh applications can ignore the resource file and access the data file exclusively.

Apple menu

Menu farthest to the left in the Macintosh menu bar; used to access the desk accessories such as Chooser.

AppleSingle

File format that stores a Macintosh file's data and resource information in separate parts of one file. Typically used for archiving files.

AppleShare

Software for AppleTalk file service developed by Apple.

AppleShare client

Computer accessing information stored on the AppleShare server.

AppleShare server

Computer used to store information for use by authorized AppleShare clients.

AppleTalk

Macintosh network protocols that allow Macintoshes to communicate with printers, file servers, and other devices over LocalTalk, Ethernet, or other cabling systems.

AppleTalk address

The network number, node number and identification number of a socket.

AppleTalk cable

See LocalTalk.

AppleTalk Link Access Protocol

See ALAP.

AppleTalk port

Serial port on an AppleTalk device used for network communication.

AppleTalk routing

GatorBox function that lets EtherTalk or TokenTalk-based Macintoshes access LaserWriters and AppleShare servers running on AppleTalk and lets LocalTalk-based Macintoshes access AppleShare servers on Ethernet. Permits AppleTalk devices in different zones to communicate.

application

Software programs that users run to perform functional tasks, such as word processing, spreadsheet analysis, and database management.

area

A set of related networks on a DECnet internet.

ARP

Address Resolution Protocol; TCP/IP protocol responsible for determining the network hardware address corresponding to a network protocol address. *Compare* RARP.

ASCII

American Standard Code for Information Interchange (pronounced ASK-ee). Code in which numbers from 0 to 255 represent individual characters, such as letters, numbers and punctuation marks; used in text representation and communications protocols.

ASP

AppleTalk Session Protocol; responsible for establishing, maintaining, and closing sessions between a user and a server.

AT

AppleTalk

atalkad

AppleTalk administration daemon that enables a UNIX-based server to administer multiple gateways on Ethernet.

ATP

AppleTalk Transaction Protocol; responsible for providing reliable transport for AppleTalk network services.

A/UX

Implementation of UNIX developed by Apple that runs on Macintosh II and SE/30 computers.

backbone

Network topology consisting of an Ethernet cable connecting two or more AppleTalk networks.

back up

(verb) To make a spare copy of a disk or file to ensure that information is retained if the original disk or file is lost or damaged.

backup

(noun) A copy of a disk or file.

back-up

(adjective) Relating to the process of backing up; for example, a back-up disk.

bandwidth

The range of transmission frequencies available on a network. Greater bandwidth means that a network can carry more information at one time.

bit

The smallest unit of information in a binary notation system; a binary digit (0 or 1).

BNC connector

Coaxial connector used with Thin Ethernet.

boot

To start a device or system.

bridge

Device connecting two or more networks of the same type, where the bridge determines which packets should be passed from one network to another. *Compare* gateway, repeater, router.

broadcast

Method of addressing nodes on network where every host hears every transmission.

button

A pushbutton-like image in Macintosh dialog boxes used to indicate that an action should be performed when the button is clicked. Bold buttons indicate a default action; you can press the Return or Enter key to accept the default.

Cancel button

Button appearing in Macintosh dialog box that cancels command or activity when clicked.

CAP

Columbia AppleTalk Package; UNIX software package that implements AppleTalk protocols on a UNIX-based server.

card

Expansion board inserted into the chassis of a computer or workstation to add memory, network interfaces, or input/output ports.

case sensitive

Able to distinguish between uppercase characters and lowercase characters.

check box

Macintosh control that indicates whether an option is on (checked) or off (unchecked). Clicking inside a check box reverses its setting.

checksum

Value used to detect transmission errors when data packets are sent from one host to another. Typically, the sending host computes the checksum for a transmission and appends the information to a packet when transmitting. The receiving host recomputes the checksum value and compares it to the value sent. If the calculated value does not match the received value, the packet is corrupted and is usually discarded.

choose

choose

To pick a command by dragging through a menu. For example, you choose the *Print* command from the File menu.

Chooser

Macintosh desk accessory that lets you control whether the AppleTalk network is active and allows you to access and use remote devices and volumes.

Chooser name

Name entered in the User Name field of the Chooser. The Chooser name identifies a Macintosh to other devices on the AppleTalk network.

Class A/B/C internet address

See IP address.

click

To position the Macintosh pointer on something and then to press and release the mouse button. *Compare* double-click, shift-click.

client

Any network node that accesses files or peripherals provided by network servers.

coaxial cable

Network wiring medium composed of a central core wire, a layer of insulation, a second wire layer, and an external insulation layer.

Command key

Macintosh key identified by clover symbol that you hold down while pressing other keys to issue commands. For example, Command-C is the keyboard equivalent to choosing Copy from the Edit menu.

config.tel file

Configuration text file used by NCSA Telnet to establish parameters governing gateways, terminal emulation, and servers.

cursor

Symbol displayed on screen to indicate where the next character typed from the keyboard will appear.

cost

Number indicating the relative efficiency of communication between two nodes on a path. A low number indicates relatively high efficiency (low cost).

daemon

A UNIX process running in the background that is responsible for a specific function or activity, such as network administration or print spooling. For example, the mountd daemon services volume mount requests.

datagram

Packet transmitted by DDP or UDP; basic unit of network transfer for connectionless networks.

DDP

Datagram Delivery Protocol; responsible for delivery of datagrams over AppleTalk networks on an internet.

default

Preset response to a question or prompt that is used by the computer if you don't supply a different response.

DESKTOP file

Invisible Macintosh resource file that identifies the name and version number of the application responsible for creating a document. The .DESKTOP file allows you to open an application by clicking on a document icon. GatorShare maintains a separate .DESKTOP file on each NFS server.

dialog box

Box displayed on a Macintosh that requests information needed to complete a command or that reports the status of a process. A dialog box is typically displayed when a menu option with an ellipsis (...) is selected.

dimmed

A menu option or dialog box item displayed in gray text rather than black text. Dimmed items are not available and cannot be selected.

directory

A file containing names and access information about other files, including other directories.

domain

A hierarchical naming convention that identifies a machine or set of machines on an internet that are administered together. A domain name is divided into subnames separated by periods. For example, in the DOC.YOYODYNE.COM domain name, COM is the top level domain and YOYODYNE and DOC identify second- and third-level subdomains.

double-click

To position the Macintosh pointer and then press and release the mouse button twice in quick succession without moving the mouse.

download server

Device from which a GatorBox obtains its configuration files and software during startup. *See* primary download server, secondary download server.

drag

To position the pointer on something, press and hold the mouse button, move the mouse, and release the mouse button.

electronic mail (e-mail)

Network service that lets users send and receive messages over a network.

encapsulation

Technique used to enclose information formatted for one protocol, such as AppleTalk, within a packet formatted for a different protocol, such as TCP/IP.

entity

Any device or process on an internet. In the AppleTalk protocols, each entity has an entity name in the form *object:type@zone*, where the *object* and *type* fields are specified by a socket client and the zone field identifies the zone in which the socket client is located. For example, CORP:LaserWriter@CORPZONE identifies the LaserWriter called CORP in the CORPZONE zone. *See* NBP.

EPROM

Erasable Programmable Read-Only Memory.

/etc/exports file

File listing the filesystems on a server that can be accessed by a remote user and the restrictions on access for each filesystem.

/etc/group table

ASCII file residing on an IP host that maps numerical group IDs to user names.

/etc/hosts file

File residing on an IP host that lists the IP addresses and names of hosts authorized to access the server.

/etc/passwd

Security file containing the login names, encrypted passwords, and other important identification for all users authorized to log in to a specific computer.

Ethernet

High-performance (10 megabits per second) cabling system developed by Xerox, Intel, and Digital Equipment Corporation that links computers and peripheral devices on a network. Sometimes used to refer to the protocols for transferring information across Ethernet cabling. *See* thin Ethernet, thick Ethernet.

EtherTalk

Protocol for sending data in AppleTalk packets over Ethernet cabling, typically by means of an Ethernet interface card. (*Compare* LocalTalk.)

export

NFS process where a server “advertises” the directories it makes available to NFS clients.

file server

Network computer that stores and manages files and applications in shared or private directories or folders.

fixed routing

See static routing.

flash EPROM

Technology used in the GatorBox CS, GatorMIM CS, and GatorBox CS/Rack that permits stable storage and easy updating of software and configuration settings.

FTP

File Transfer Protocol; standard high-level internet protocol for transferring files from one host to another.

gateway

Hardware and software that connect networks that use different protocols, such as AppleTalk and TCP/IP. The gateway translates between the protocols so that devices on the connected networks can exchange data. *Compare* bridge, repeater, router.

GatorBox

Intelligent network gateway that integrates Macintosh computers into Ethernet network environments.

GatorBoxName

The name assigned to a GatorBox. A GatorBox is initially assigned its five- or six-digit serial number as part of its name; for example, "GatorBox100019."

GatorCard

Ethernet card installed in a Macintosh SE, Macintosh SE/30, or Macintosh II-type computer that allows it to connect directly to an Ethernet network.

GatorDatabase

File containing the list of servers accessible to a GatorBox.

GatorKeeper

Administration program for the GatorBox that lets a user configure and monitor the GatorBox gateway functions.

GatorMIM CS

Media interface module version of the GatorBox CS designed to fit in a Cabletron Multi-Media Access Center (MMAC).

GatorPrint

Cayman's UNIX-to-LocalTalk software for the GatorBox. GatorPrint also provides AppleTalk routing and TCP/IP services. *Compare* GatorShare, GatorSystem.

GatorShare

Cayman's file-sharing software for the GatorBox that provides transparent access to NFS file servers, including Sun, VAX, NeXT, Apollo, and Macintoshes running A/UX. GatorShare also provides AppleTalk routing and TCP/IP services. *Compare* GatorSystem, GatorPrint.

GatorSystem

GatorBox software responsible for AppleTalk routing and TCP/IP services. *Compare* GatorPrint, GatorShare.

group

The set of users who are assigned the same access privileges for a file or directory. A user can belong to multiple groups.

hard routing

See static routing.

hop

The logical distance between two nodes on an internet. If a packet traveling from one node to another must pass through two routers, the path has a hop count of 2.

host

A computer on a network that acts as a central processing unit for one or more end users. A host can be a personal computer used by one person or a mainframe computer with hundreds of terminal connections.

ICMP

Internet Control Message Protocol; IP protocol responsible for reporting delivery or routing problems to a sending gateway or host.

internet

A network created by linking two or more smaller networks of the same or different types with a router or gateway.

internet address

See IP address.

IP

internet Protocol; network-layer protocol responsible for directing information packets from one computer to another over an internet. IP accepts data in segments, encapsulates the data in packets, and determines the correct path for routing the packet to its destination.

IP address

4-byte identification code assigned to each device on a TCP/IP internet. The IP address is divided into a network segment, which identifies a network on the internet, and a host segment, which identifies a host attached to the specified network. IP addresses are divided into three classes:

- ▶ **Class A** — Address structure that supports networks with more than 16 million nodes. The first node of a Class A address is a number in the range 0-126.
- ▶ **Class B** — Address structure that supports networks with up to 65,536 nodes. The first node of a Class B address is a number in the range 128-191.
- ▶ **Class C** — Address structure that supports networks with up to 256 nodes. The first node of a Class C address is a number in the range 192-223.

ISO

International Standards Organization; developer of the OSI reference model for network standardization. *See* OSI.

KIP

Protocol that encapsulates AppleTalk packets in UDP/IP packets. KIP lets the GatorBox give Macintoshes on LocalTalk or EtherTalk access to IP-based computers that understand AppleTalk protocols.

LAN

Local area network; network of computers and peripherals connected by cabling within a specific area, such as a building or a section of a building. *Compare* WAN.

LAP

Link Access Protocol. *See* ALAP.

level 1 router (DECnet)

A DECnet router that can forward packets from one network to another in the same area.

level 2 router (DECnet)

A DECnet router that can forward packets from one network to another in the same area or in different areas.

LocalTalk

Shielded twisted-pair cabling system operating at 230 kilobits per second that supports as many as 32 nodes (computers or peripheral devices) on a network. One LocalTalk connection box is required for each computer and peripheral on the network.

LocalTalk network

Network using AppleTalk protocols over LocalTalk cabling. *See* EtherTalk network.

local volume

Volume to which the user is connected. *Compare* remote volume.

log in

To gain access to a server or host as an authorized user.

log out

To terminate a connection to a server or host.

MacIP

GatorBox function that supports Macintosh networking applications that encapsulate IP protocol packets inside AppleTalk packets. This encapsulation lets Macintoshes on AppleTalk networks communicate with Ethernet-based computers that support TCP/IP.

MacTCP

Apple's implementation of the TCP/IP protocol suite. A Macintosh with MacTCP installed in its System folder can connect to an internet and can access other computers and networks using TCP/IP.

mail server

Network computer that stores messages sent from one mail client to another.

media

The physical conductor of network transmissions. Examples include Ethernet and LocalTalk.

mount point

A directory on a file system “exported” for use by remote NFS clients.

mount protocol

Procedure by which a remote volume is made accessible to a user.

mounted volume

Volume to which a user has gained access over a network. A mounted volume can be either local (that is, physically connected to the Macintosh) or remote (that is, connected to the Macintosh over a network or internet).

name server

Network computer that maps host names to IP addresses, simplifying network administration.

NBP

Name Binding Protocol; AppleTalk protocol responsible for translating an object name (*object:type@zone*) into an internet address. An *NBP lookup* command requests information about all devices with a specified name or type or in a specified zone.

NCSA Telnet

Terminal emulation and file transfer program for the Macintosh developed by the National Center for Supercomputing Applications and distributed by Cayman Systems with the GatorBox. NCSA Telnet handles translation between a LocalTalk-based or EtherTalk-based Macintosh and a Telnet host on networks using TCP/IP.

network

Group of interconnected devices, including file servers, host computers, and peripherals.

Network Information Service

Distributed database service typically used to distribute user and group access information in authentication databases around a network. Formerly called “Yellow Pages.”

network number

Integer that identifies a network to other network servers. Each AppleTalk network on an internet must have a unique network number in the range 1-65534.

NFS

Network File System; a file-sharing protocol developed by Sun Microsystems that has been adopted as a standard by vendors of workstations and minicomputers.

NIC

Network Information Center; responsible for the global assignment of IP network numbers.

node

An individual computer or peripheral that has an address on a network.

node number

The unique identifier for a node on a network.

NVRAM

Nonvolatile random access memory; component of the GatorBox that stores download server address information while the GatorBox is turned off.

OSI

Open Systems Interconnection; a seven-layer reference model that creates a framework for network standards and protocols:

7. **Application** — Defines the interface of application software with the network's operating system.
6. **Presentation** — Reformats differences in user data into a format usable by the network's application layer.
5. **Session** — Used for administrative tasks, such as security.
4. **Transport** — Provides error checking and routing of data packets.
3. **Network** — Provides switching and routing rules.
2. **Data link** — Controls data flow in and out of network devices.
1. **Physical** — Controls physical network connection of workstations to the wiring media.

packet

Formatted unit of information sent as a discrete string of bits over network from one node to another. Information is separated into multiple packets before being transferred.

pathname

The full name by which an operating system identifies a file. A pathname is a sequence of filenames, each preceded by a separator character, such as a slash, that specifies the path—from volume directory to file—the operating system takes to locate that file.

PAP

Printer Access Protocol.

PC AppleShare

Application run on an IBM personal computer or compatible that enables the computer to connect to an AppleTalk network.

permissions

File attributes that specify whether the owner or group of the file and/or the public can read, write, or execute a file (or directory).

PhoneNET

Network cabling system developed by Farallon Computing that uses telephone wiring instead of LocalTalk cabling as the AppleTalk medium.

port

(1) Socket on the back panel of a computer or GatorBox where you can plug in a cable to connect a peripheral device, another computer, or a network. For example, the Macintosh printer port lets you connect the Macintosh to a printer or AppleTalk connector. (2) An end point in a connection between TCP/IP hosts.

print server

Networked computer that lets workstations submit print requests to it simultaneously.

process

UNIX program that provides a specific function to an end user or a client program.

protocol

Formal set of rules that govern the transmission of information across a network.

power-cycle

To turn a device off and on.

radio button

Macintosh control that displays a setting, such as "Off" or "On." Radio buttons typically appear as part of a group, where only one button can be on at a time: when you click one button in a group, the other buttons in the group are turned off.

RAM

Random access memory.

RARP

Reverse Address Resolution Protocol; responsible for determining the internet address corresponding to an Ethernet network hardware address. *Compare* ARP.

remote volume

Volume connected to a Macintosh over a network or internet. *Compare* local volume.

repeater

Device that connects two networks of the same type, where the repeater amplifies and transfers electrical signals from one segment of a network to another segment of the same network automatically. *Compare* bridge, router, gateway.

retry count

The number of times a GatorBox will resend unanswered packets to a server.

retry interval

Time interval between attempts to send packets. Specified by client.

RFC

Request For Comment; set of documents that establish standards for TCP/IP networking.

RIP

Routing Information Protocol; protocol used by IP routers to send and receive network routing information.

ROM

Read-only memory.

root

Login identification code of anyone holding superuser privileges, such as update access to the password file and system administration files in /etc.

router

Device connecting two or more networks of the same type that routes packets from one network to another based on the packet destination address. *Compare* bridge, gateway, repeater.

routing

Process by which network packets are transmitted from one network to another across a gateway or internet.

RPC

Remote Procedure Call; protocol that enables communications over a network between processes running on different machines.

RTMP

Routing Table Maintenance Protocol; protocol used by AppleTalk routers to update their routing databases.

server

Any network computer or device that provides services for other devices (clients) on the network.

shielded cable

Cable with metallic wrap around the wires to reduce potential effects of radio-frequency interference.

shift-click

To select multiple objects by clicking on each object while holding down the Shift key. *Compare* click, double-click.

SMTP

Simple Mail Transfer Protocol; protocol responsible for transferring electronic mail from one host to another across an internet.

SNMP

Simple Network Management Protocol; protocol that allows a network administrator to monitor network devices over the network.

socket

Logical entity within a node on an AppleTalk or IP network. On a stand-alone AppleTalk network, each socket is identified by its AppleTalk address (node ID and socket number). On an internet, each socket is identified by its internet address (network number, node ID, and socket number).

standard Ethernet

See thick Ethernet.

static routing

Sending (or tunneling) AppleTalk or EtherTalk packets from one AppleTalk network through an IP network to a second AppleTalk network. Routing is "static" because routes are not determined by means of a routing information protocol.

subnet

Networking scheme that separates one logical IP network into multiple smaller logical networks to simplify maintenance and packet routing.

subnet mask

Code indicating the portion of an IP address reserved for a subnetwork identifier and the portion reserved for a host identifier. A subnet mask bit is set to 1 if the corresponding bit in the IP address is part of the network number and subnetwork network fields, and to 0 if the corresponding bit in the address is part of the host number field.

superuser (su)

User ID of the System Administrator. Modification of important system files requires superuser access.

System file

Macintosh system software file that provides system-wide information and resources. The System file must be in the System folder.

System Folder

Folder containing the System and Finder files. GatorKeeper looks in the System Folder for the GatorBox software and configuration information if it cannot find them in its own folder.

TCP

Transmission Control Protocol; protocol for error-free transmission and reception of datagrams over an IP network.

TCP/IP

Transmission Control Protocol/Internet Protocol.

Telnet

See NCSA Telnet.

terminal emulation

Program function that allows a Macintosh to log in to a host computer system as a specific type of data terminal.

text file

File containing only unformatted text (typically ASCII).

TFTP

Trivial File Transfer Protocol; protocol responsible for transferring files, such as the GatorDatabase, from the TFTP server (Macintosh or remote server) to the TFTP client (GatorBox).

TFTP server

Macintosh or IP server used to download configuration information to a GatorBox.

thick Ethernet

Ethernet cable 0.5" in diameter.

thin Ethernet

Ethernet cable distinguished by its flexibility and economy.

tn3270

Application developed at Brown University that allows a Macintosh to emulate an IBM 3270 terminal.

Token Ring

Cabling system that links computers and peripheral devices on a continuous network loop. Each host on a Token Ring network passes an electronic *token*, which lets it transmit data on the network or pass the token to the next host.

TokenTalk

AppleTalk protocols running over Token Ring cabling.

transceiver

Device required to connect hosts or devices to thick Ethernet cabling.

transceiver Ethernet

See thick Ethernet.

twisted pair cable

Network cabling consisting of two insulated wires wrapped around each other.

UDP

User Datagram Protocol; transport-layer protocol that has lower protocol overhead and less reliability than TCP in the transport of data between host programs.

UNIX

A multi-user, multi-tasking operating system that functions on a variety of mainframe and personal computers.

volume

Storage device formatted to contain files. A volume can be a physical or virtual disk, and typically appears as a separate icon on a user's workstation. *See* local volume, remote volume.

WAN

Wide area network; network of computers and peripherals connected over large areas by means of cabling or other connections. *Compare* LAN.

window

Object on the Macintosh desktop that presents information, such as a document or message. Windows can typically be moved and resized.

Yellow Pages

See Network Information Service.

YP

See Network Information Service.

ZIP

Zone Information Protocol; AppleTalk protocol responsible for maintaining an internet-wide mapping of networks to zone names.

zone

A subset of the AppleTalk (that is, LocalTalk and EtherTalk) networks on an internet. A zone can consist of a single AppleTalk network or several AppleTalk networks.

zone name

The name of one or more AppleTalk zones on your network. Zone names can be viewed by using the Chooser.

Appendix B

GatorBox SNMP MIB

The following appendix lists Cayman Systems's private SNMP Management Information Base (MIB). The most current version of the GatorBox MIB is available via anonymous ftp from <ftp.cayman.com>.

```
--
-- Cayman GatorBox AppleTalk MIB
--
--
-- 1.  This version of the Cayman private AppleTalk MIB is a duplicate of
--      the AppleTalk MIB in the Kinetics FastPath KStar V8.0 software.
--      This duplication of private MIB variable definitions is done as a
--      courtesy to our customers.
--
--      FASTPATH-MIB { private(4) enterprises(1) excelan(23)
--                      mibDoc(2) fastpath(11) }
--
-- 2.  This is the first release of an RFC1066 MIB with RFC1067 SNMP. It is
--      not yet fully compliant. It is, however, functional and useful as a
--      monitoring tool.
--
--      The RFC1066 MIB for TCP/IP is also supported with the following
--      exceptions:
--
-- History:
--      Originally created by Steve Waldbusser at Carnegie Mellon
--
--      v1.0    11/89    initially formed from Kinetics KStar V8.0 MIB
--
--              02/89    cleaned up some syntax errors; changed to be
--              parsable by CMU ASN.1 parser.
--
--              05/89    commented out bogus DEFINITION lines.
--
```



```

-- Notes:
--
-- The tcp variables are present but are unsupported; While TCP is present,
-- the SNMP MIB variables are not yet supported.
-- The egg variables are present but are unsupported; There is no protocol
-- support for EGP in the GatorBox.
-- The may be slight inconsistencies about reporting of some variables as
-- we have not yet completed a full review of the implementation to
-- determine compliance.
--
-- scc, alap and ethernet are prefixed by 2 byte interface # ?
--
-- atifTable is indexed by trailing 1 byte entry number
-- rtmpTable is indexed by trailing 2 byte network number
-- zipTable is indexed by trailing 1 byte entry number

-- CAYMAN-MIB { private(4) enterprises(1) cayman(7) gatorbox(1) }
--
-- DEFINITIONS ::= BEGIN
--
-- IMPORTS
--
--         OBJECT-TYPE, NetworkAddress, IPAddress,
--         Counter, Gauge, TimeTicks
--
-- FROM RFC1065-SMI;

cayman          OBJECT IDENTIFIER ::= { enterprises 7 }
gatorbox        OBJECT IDENTIFIER ::= { cayman 1 }
gatorbox-id     OBJECT IDENTIFIER ::= { cayman 2 }
scc             OBJECT IDENTIFIER ::= { gatorbox 1 }
alap           OBJECT IDENTIFIER ::= { gatorbox 2 }
ethernet        OBJECT IDENTIFIER ::= { gatorbox 3 }
aarp           OBJECT IDENTIFIER ::= { gatorbox 4 }
atif           OBJECT IDENTIFIER ::= { gatorbox 5 }
ddp            OBJECT IDENTIFIER ::= { gatorbox 6 }
rtmp           OBJECT IDENTIFIER ::= { gatorbox 7 }
kip            OBJECT IDENTIFIER ::= { gatorbox 8 }
zip            OBJECT IDENTIFIER ::= { gatorbox 9 }
nbp            OBJECT IDENTIFIER ::= { gatorbox 10 }
echo           OBJECT IDENTIFIER ::= { gatorbox 11 }

```

```

buffer          OBJECT IDENTIFIER ::= { gatorbox 12 }

-- Product id

name            OBJECT-TYPE
SYNTAX          OCTET STRING
-- DEFINITION   The name of the product in human-readable form.
ACCESS          read-only
STATUS          mandatory
::= { gatorbox-id 1 }

-- The SCC Group

sccInterruptCount OBJECT-TYPE
SYNTAX          Counter
-- DEFINITION   The total number of receive interrupts on this
--             interface.
ACCESS          read-only
STATUS          mandatory
::= { scc 1 }

sccAbortCount   OBJECT-TYPE
SYNTAX          Counter
-- DEFINITION   The total number of abort interrupts on this
--             interface.
ACCESS          read-only
STATUS          mandatory
::= { scc 2 }

sccSpuriousCount OBJECT-TYPE
SYNTAX          Counter
-- DEFINITION   The total number of spurious interrupts on this
--             interface.
ACCESS          read-only
STATUS          mandatory
::= { scc 3 }

```

sccCRCCount OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of CRC errors on this interface.

ACCESS read-only

STATUS mandatory

::= { scc 4 }

sccOverrunCount OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of receive overrun errors on this
-- interface.

ACCESS read-only

STATUS mandatory

::= { scc 5 }

sccUnderrunCount OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of receive underrun errors on this
-- interface.

ACCESS read-only

STATUS mandatory

::= { scc 6 }

The ALAP Group

OBJECT-TYPE SYNTAX DEFINITION ACCESS STATUS

alapReceiveCount Counter The total number of good packets received
on this LocalTalk interface. R/O mand

{ alap 1 }

alapTransmitCount Counter

-- DEFINITION The total number of packets transmitted on this
-- LocalTalk interface.

ACCESS read-only

STATUS mandatory

::= { alap 2 }

alapNoHandlerCount OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of packets received on this
-- LocalTalk interface for which there was no protocol
-- handler.
ACCESS read-only
STATUS mandatory
::= { alap 3 }

alapLengthErrorCount OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of packets received on this
-- LocalTalk interface whose actual length did not match
-- the length in its header.
ACCESS read-only
STATUS mandatory
::= { alap 4 }

alapBadCount OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of bad packets received on this
-- LocalTalk interface.
ACCESS read-only
STATUS mandatory
::= { alap 5 }

alapCollisionCount OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of collisions assumed on this
-- LocalTalk interface.
ACCESS read-only
STATUS mandatory
::= { alap 6 }

```

alapDeferCount OBJECT-TYPE
SYNTAX          Counter
-- DEFINITION   The total number of times this LocalTalk interface
--              deferred to other packets.
ACCESS          read-only
STATUS          mandatory
 ::= { alap 7 }

```

```

alapNoDataCount OBJECT-TYPE
SYNTAX          Counter
-- DEFINITION   The total number of times this LocalTalk interface
--              received an RTS packet and expected a data packet,
--              but did not receive any data packet.
ACCESS          read-only
STATUS          mandatory
 ::= { alap 8 }

```

```

alapRandomCTS OBJECT-TYPE
SYNTAX          Counter
-- DEFINITION   The total number of times this LocalTalk interface
--              received a CTS packet that was not solicited by an
--              RTS packet.
ACCESS          read-only
STATUS          mandatory
 ::= { alap 9 }

```

```

-- The Ethernet Group

```

```

etherCRCErrors OBJECT-TYPE
SYNTAX          Counter
-- DEFINITION   The total number of CRC errors on this ethernet
--              interface.
ACCESS          read-only
STATUS          mandatory
 ::= { ethernet 1 }

```

etherAlignErrors OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of alignment errors on this ethernet
-- interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 2 }

etherResourceError OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of errors due to lack of resources
-- on thick ethernet interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 3 }

etherOverrunErrors OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of overrun errors on this ethernet
-- interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 4 }

etherInPackets OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of input packets on this ethernet
-- interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 5 }

etherOutPackets OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of output packets on this ethernet
-- interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 6 }

etherBadTransmits OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of transmission errors on this
-- ethernet interface.
ACCESS read-only
STATUS mandatory
::= { ethernet 7 }

etherOversizeFrames OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of oversize frame errors on this
-- ethernet interface.
ACCESS read-only
STATUS mandatory
::= { ethernet 8 }

etherSpurRUReadys OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of spurious RU Ready interrupts on
-- this ethernet interface.
ACCESS read-only
STATUS mandatory
::= { ethernet 9 }

etherSpurCUActives OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of spurious CU Active interrupts on
-- this ethernet interface.
ACCESS read-only
STATUS mandatory
::= { ethernet 10 }

etherSpurUnknown OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of unknown spurious interrupts on
-- this ethernet interface.
ACCESS read-only
STATUS mandatory
::= { ethernet 11 }

etherBcastDrops OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of broadcast packets dropped to free
-- resources on this ethernet interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 12 }

etherReceiverRestarts OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of receiver restarts on this
-- ethernet interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 13 }

etherReinterrupts OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of reinterrupts on this ethernet
-- interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 14 }

etherBufferReroutes OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of buffers taken off of queues to
-- service incoming packets on this ethernet interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 15 }

etherBufferDrops OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of buffers dropped on this ethernet
-- interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 16 }

etherCollisions OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of collisions encountered on this
-- ethernet interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 17 }

etherDefers OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of deferrals encountered on this
-- ethernet interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 18 }

etherDMAUnderruns OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of DMA Underruns on this ethernet
-- interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 19 }

etherMaxCollisions OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of packets dropped on this ethernet
-- interface because they encountered more than 16
-- collisions.

ACCESS read-only

STATUS mandatory

::= { ethernet 20 }

etherNoCarriers OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of no carrier errors experienced on
-- this ethernet interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 21 }

etherNoCTS OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of no CTS errors experienced on this
-- ethernet interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 22 }

etherNoSQEs OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of no SQE errors experienced on this
-- ethernet interface.

ACCESS read-only

STATUS mandatory

::= { ethernet 23 }

-- The AARP Group

aarpTable OBJECT-TYPE

SYNTAX SEQUENCE OF AarpEntry

-- DEFINITION The AppleTalk Address Resolution Table contains an
-- equivalence of AppleTalk Network Addresses to the
-- link layer "physical" address.

ACCESS read-write

STATUS mandatory

::= { aarp 1 }

aarpEntry OBJECT-TYPE

SYNTAX SEQUENCE OF AarpEntry

-- DEFINITION Each entry contains one AppleTalk Network Address to
-- "physical" address equivalence.

ACCESS read-write

STATUS mandatory

::= { aarpTable 1 }

aarpEntry ::= SEQUENCE {

aarpIfIndex INTEGER,

aarpPhysAddress OCTET STRING,

aarpNetAddress OCTET STRING

}

aarpIfIndex OBJECT-TYPE

SYNTAX INTEGER

-- DEFINITION The interface on which this entry's equivalence is
-- effective; The interface identified by a particular
-- value of this index is the same interface as
-- identified by the same value of ifIndex.

ACCESS read-only

STATUS mandatory

::= { aarpEntry 1 }

aarpPhysAddress OBJECT-TYPE

SYNTAX OCTET STRING

-- DEFINITION The media-dependent "physical address".

ACCESS read-only

STATUS mandatory

::= { aarpEntry 2 }

aarpNetAddress OBJECT-TYPE

SYNTAX OCTET STRING

-- DEFINITION The AppleTalk Network Address corresponding to the
-- media-dependent "physical address".

ACCESS read-only

STATUS mandatory

::= { aarpEntry 3 }

atifTable OBJECT-TYPE

SYNTAX SEQUENCE OF AtifEntry

-- DEFINITION The description of one of the logical interfaces on
-- this entity.

ACCESS read-write

STATUS mandatory

::= { atif 1 }

atifEntry OBJECT-TYPE

SYNTAX AtifEntry

-- DEFINITION The description of one of the logical appletalk
-- interfaces on this entity.

ACCESS read-write

STATUS mandatory

::= { atifTable 1 }

atifEntry ::= SEQUENCE {

 atifIndex INTEGER,
 atifDescr OCTET STRING,
 atifType INTEGER,
 atifNetStart OCTET STRING,
 atifNetEnd OCTET STRING,
 atifNetAddress OCTET STRING
 atifStatus INTEGER,
 atifNetConfig INTEGER,
 atifZoneConfig INTEGER,
 atifZone OCTET STRING,
 atifIfIndex INTEGER

}

atifIndex OBJECT-TYPE

SYNTAX INTEGER

-- DEFINITION A unique value for each logical AppleTalk interface;
-- Its value is between 1 and the total number of
-- logical AppleTalk interfaces.

ACCESS read-only

STATUS mandatory

::= { atifEntry 1 }

atifDescr OBJECT-TYPE

SYNTAX OCTET STRING

-- DEFINITION A text string containing information about the
-- interface; This string is intended for presentation
-- to a human; it must not contain anything but
-- printable ASCII characters.

ACCESS read-only

STATUS mandatory

::= { atifEntry 2 }

atifType OBJECT-TYPE

SYNTAX INTEGER {

other(1), -- none of the following
localtalk(2),
ethertalk1(3),
ethertalk2(4),
tokentalk(5),
iptalk(6)

}

-- DEFINITION The type of interface, distinguished by the protocol
-- immediately "below" DDP in the protocol stack.

ACCESS read-only

STATUS mandatory

::= { atifEntry 3 }

atifNetStart OBJECT-TYPE

SYNTAX OCTET STRING

-- DEFINITION The first Appletalk network address in the range
-- configured for this interface, or the single network
-- number configured for this interface.

ACCESS read-write

STATUS mandatory

::= { atifEntry 4 }

atifNetEnd OBJECT-TYPE

SYNTAX OCTET STRING

-- DEFINITION The last Appletalk network address in the range
-- configured for this interface, or zero if a single
-- network number is configured.

ACCESS read-write

STATUS mandatory

::= { atifEntry 5 }

atifNetAddress OBJECT-TYPE

SYNTAX OCTET STRING

-- DEFINITION The Appletalk network address configured for this interface.

ACCESS read-write

STATUS mandatory

::= { atifEntry 6 }

atifStatus OBJECT-TYPE

SYNTAX INTEGER {
 operational(1),
 unconfigured(2),
 off(3)
 }

-- DEFINITION The configuration status of this interface.

ACCESS read-only

STATUS mandatory

::= { atifEntry 7 }

atifNetConfig OBJECT-TYPE

SYNTAX INTEGER {
 configured(1),
 garnered(2),
 guessed(3),
 unconfigured(4)
 }

-- DEFINITION The configuration status of the DDP network number(s)
-- for this interface.

ACCESS read-only

STATUS mandatory

::= { atifEntry 8 }

atifZoneConfig OBJECT-TYPE

SYNTAX INTEGER {
 configured(1),
 garnered(2),
 guessed(3),
 unconfigured(4)
 }

-- DEFINITION The configuration status of AppleTalk zone name(s)
-- for this interface.

ACCESS read-only

STATUS mandatory

::= { atifEntry 9 }

atifZone OBJECT-TYPE

SYNTAX OCTET STRING

-- DEFINITION The zone name configured for this logical interface;
-- In Phase 2 networks, this is the "default" zone for
-- this interface.

ACCESS read-write

STATUS mandatory

::= { atifEntry 10 }

atifIfIndex OBJECT-TYPE

SYNTAX INTEGER

-- DEFINITION The physical interface associated with this logical
-- interface; A particular value of this index
-- identifies the same interface as is identified by the
-- same value of ifIndex.

ACCESS read-write

STATUS mandatory

::= { atifEntry 11 }

-- The DDP Group

ddpOutRequests OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of DDP datagrams which were supplied
-- to DDP in requests for transmission.

ACCESS read-only

STATUS mandatory

::= { ddp 1 }

ddpOutShort OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of short DDP datagrams which were
-- transmitted from this entity.

ACCESS read-only

STATUS mandatory

::= { ddp 2 }

ddpOutLong OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of long DDP datagrams which were
-- transmitted from this entity.

ACCESS read-only

STATUS mandatory

::= { ddp 3 }

ddpReceived OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of input datagrams received by DDP.
ACCESS read-only
STATUS mandatory
::= { ddp 4 }

ddpToForward OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The number of input datagrams for which this entity
-- was not their final DDP destination, as a result of
-- which an attempt was made to find a route to forward
-- them to that final destination.
ACCESS read-only
STATUS mandatory
::= { ddp 5 }

ddpForwards OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The number of input datagrams for which this entity
-- was not their final DDP destination, as a result of
-- which they were forwarded to their final destination.
ACCESS read-only
STATUS mandatory
::= { ddp 6 }

ddpForMe OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of input DDP datagrams for which
-- this entity was their final DDP destination.
ACCESS read-only
STATUS mandatory
::= { ddp 7 }

ddpNoProtocolHandler OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of DDP datagrams addressed to this
-- entity that were addressed to an upper layer protocol
-- for which no protocol handler existed.
ACCESS read-only
STATUS mandatory
::= { ddp 8 }

ddpOutNoRoutes OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of DDP datagrams dropped because a
-- route could not be found to their final destination.
ACCESS read-only
STATUS mandatory
::= { ddp 9 }

ddpTooShortDrops OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of input DDP datagrams dropped
-- because the received data length was less than the
-- data length in the DDP header.
ACCESS read-only
STATUS mandatory
::= { ddp 10 }

ddpTooLongDrops OBJECT-TYPE

SYNTAX Counter
-- DEFINITION The total number of input DDP datagrams dropped
-- because they exceeded the maximum DDP datagram size
-- or because their header size was greater than their
-- length.
ACCESS read-only
STATUS mandatory
::= { ddp 11 }

ddpBroadcastDrops OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of input DDP datagrams dropped
-- because this entity was not their final destination
-- and they were addressed to the link level broadcast.

ACCESS read-only

STATUS mandatory

::= { ddp 12 }

ddpShortDDPDrops OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of input DDP datagrams dropped
-- because this entity was not their final destination
-- and their type was short DDP.

ACCESS read-only

STATUS mandatory

::= { ddp 13 }

ddpHopCountDrops OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of input DDP datagrams dropped
-- because this entity was not their final destination
-- and their hop count would exceed 16.

ACCESS read-only

STATUS mandatory

::= { ddp 14 }

-- The RTMP Group

rtmpTable OBJECT-TYPE

SYNTAX SEQUENCE OF RtmpEntry

-- DEFINITION This entity's RTMP table.

ACCESS read-write

STATUS mandatory

::= { rtmp 1 }

```

rtmpEntry OBJECT-TYPE
SYNTAX      RtmpEntry
-- DEFINITION The route entry to a particular range of networks.
ACCESS      read-write
STATUS      mandatory
::= { rtmpTable 1 }

```

```

RtmpEntry ::= SEQUENCE {
    rtmpRangeStart  OCTET STRING,
    rtmpRangeEnd    OCTET STRING,
    rtmpNextHop     OCTET STRING,
    rtmpInterface   INTEGER,
    rtmpHops        INTEGER,
    rtmpState       INTEGER
}

```

```

rtmpRangeStart OBJECT-TYPE
SYNTAX      OCTET STRING
-- DEFINITION The first DDP network address in the range of
--            networks that this routing entry pertains to, or the
--            single network number that this routing entry
--            pertains to; This is a two octet DDP network
--            address.
ACCESS      read-write
STATUS      mandatory
::= { rtmpEntry 1 }

```

```

rtmpRangeEnd OBJECT-TYPE
SYNTAX      OCTET STRING
-- DEFINITION The last DDP network address in the range of networks
--            that this routing entry pertains to, or zero if this
--            routing entry pertains to a single network number;
--            This is a two octet DDP network address.
ACCESS      read-write
STATUS      mandatory
::= { rtmpEntry 2 }

```

rtmpNextHop OBJECT-TYPE

SYNTAX OCTET STRING

-- DEFINITION The next hop in the route to this entry's destination
-- network.

ACCESS read-write

STATUS mandatory

::= { rtmpEntry 3 }

rtmpInterface OBJECT-TYPE

SYNTAX INTEGER

-- DEFINITION The index of the logical appletalk interface over
-- which this route points.

ACCESS read-write

STATUS mandatory

::= { rtmpEntry 4 }

rtmpHops OBJECT-TYPE

SYNTAX INTEGER

-- DEFINITION The number of hops required to reach the destination
-- network that this entry pertains to.

ACCESS read-write

STATUS mandatory

::= { rtmpEntry 5 }

rtmpState OBJECT-TYPE

SYNTAX INTEGER {
 good(1),
 suspect(2),
 bad(3)
 }

-- DEFINITION The status of the information contained in this route entry.

ACCESS read-write

STATUS mandatory

::= { rtmpEntry 6 }

-- The KIP Group

kipTable OBJECT-TYPE

SYNTAX SEQUENCE OF KipEntry

-- DEFINITION The table of routing information for KIP networks.

ACCESS read-write

STATUS mandatory

::= { kip 1 }

kipEntry OBJECT-TYPE

SYNTAX KipEntry

-- DEFINITION An entry in the routing table for RIP networks.

ACCESS read-write

STATUS mandatory

::= { kipTable 1 }

KipEntry ::= SEQUENCE {

kipNet OCTET STRING,

kipNextHop IPADDRESS,

kipHopCount INTEGER,

kipBCastAddr IPADDRESS,

kipCore INTEGER,

kipKfps INTEGER

}

kipNet OBJECT-TYPE

SYNTAX OCTET STRING

-- DEFINITION The appletalk network address for this routing entry.

ACCESS read-write

STATUS mandatory

::= { kipEntry 1 }

kipNextHop OBJECT-TYPE

SYNTAX IPADDRESS

-- DEFINITION The IP address of the next hop for this routing entry.

ACCESS read-write

STATUS mandatory

::= { kipEntry 2 }

```

kipHopCount OBJECT-TYPE
SYNTAX          INTEGER
-- DEFINITION   The distance in hops to the destination of this route.
ACCESS          read-write
STATUS          mandatory
::= { kipEntry 3 }

kipBCastAddr OBJECT-TYPE
SYNTAX          IPADDRESS
-- DEFINITION   The form of the IP address used to broadcast on this network.
ACCESS          read-write
STATUS          mandatory
::= { kipEntry 4 }

kipCore OBJECT-TYPE
SYNTAX          INTEGER {
                    core(1),
                    notcore(2)
                }
-- DEFINITION   The status of this network as a Core network.
ACCESS          read-write
STATUS          mandatory
::= { kipEntry 5 }

kipKfps OBJECT-TYPE
SYNTAX          INTEGER {
                    kfps(1),
                    notkfps(2)
                }
-- DEFINITION   The type of the device that this network resides on.
ACCESS          read-write
STATUS          mandatory
::= { kipEntry 6 }

```

-- The ZIP Group

zipTable OBJECT-TYPE

SYNTAX SEQUENCE OF ZipEntry

-- DEFINITION The table of zone information for reachable AppleTalk
-- networks.

ACCESS read-write

STATUS mandatory

::= { zip 1 }

zipEntry OBJECT-TYPE

SYNTAX ZipEntry

-- DEFINITION An entry of zone information for a particular zone
-- and network combination.

ACCESS read-write

STATUS mandatory

::= { zipTable 1 }

ZipEntry ::= SEQUENCE {

zipZoneName OCTET STRING,

zipZoneIndex INTEGER,

zipNetStart OCTET STRING,

zipNetEnd OCTET STRING

}

zipZoneName OBJECT-TYPE

SYNTAX OCTET STRING

-- DEFINITION The ASCII zone name of this entry.

ACCESS read-write

STATUS mandatory

::= { zipEntry 1 }

zipZoneIndex OBJECT-TYPE

SYNTAX INTEGER

-- DEFINITION An integer that is unique to the zipZoneName that is
-- present in this entry; For any given zone name,
-- every zipEntry that has that zone name will have the
-- same zipZoneIndex (in a given GatorBox, as long as it
-- has not been rebooted).

ACCESS read-only

STATUS mandatory

::= { zipEntry 2 }

zipZoneNetStart OBJECT-TYPE

SYNTAX OCTET STRING

-- DEFINITION The network that starts the range for this entry, or
-- the single network number for this entry.

ACCESS read-write

STATUS mandatory

::= { zipEntry 3 }

zipZoneNetEnd OBJECT-TYPE

SYNTAX OCTET STRING

-- DEFINITION The network that ends the range for this entry, or
-- zero if this entry refers to a single network number.

ACCESS read-write

STATUS mandatory

::= { zipEntry 4 }

-- The NBP Group

nbpTable OBJECT-TYPE

SYNTAX SEQUENCE OF NbpEntry

-- DEFINITION The table of NBP services registered on this entity.

ACCESS read-write

STATUS mandatory

::= { nbp 1 }

```

nbpEntry OBJECT-TYPE
SYNTAX      NbpEntry
-- DEFINITION The description of an NBP service registered on this
--            entity.
ACCESS      read-write
STATUS      mandatory
::= { nbpTable 1 }

```

```

nbpEntry ::= SEQUENCE {
    nbpIndex      INTEGER,
    nbpObject      OCTET STRING,
    nbpType      OCTET STRING,
    nbpZone      OCTET STRING
}

```

```

nbpIndex OBJECT-TYPE
SYNTAX      INTEGER
-- DEFINITION The index of this NBP entry; this value ranges from
--            1 to the number of NBP entries registered on this
--            entity.
ACCESS      read-only
STATUS      mandatory
::= { nbpEntry 1 }

```

```

nbpObject OBJECT-TYPE
SYNTAX      OCTET STRING
-- DEFINITION The name of the service described by this entity.
ACCESS      read-write
STATUS      mandatory
::= { nbpEntry 2 }

```

```

nbpType OBJECT-TYPE
SYNTAX      OCTET STRING
-- DEFINITION The type of the service described by this entity.
ACCESS      read-write
STATUS      mandatory
::= { nbpEntry 3 }

```

```

nbpZone OBJECT-TYPE
SYNTAX      OCTET STRING
-- DEFINITION The zone the service described by this entity is
--            registered in.
ACCESS      read-write
STATUS      mandatory
::= { nbpEntry 4 }

```

-- The Echo Group

```

echoRequests OBJECT-TYPE
SYNTAX      Counter
-- DEFINITION The number of echo requests received.
ACCESS      read-only
STATUS      mandatory
::= { echo 1 }

```

```

echoReplies OBJECT-TYPE
SYNTAX      Counter
-- DEFINITION The number of echo replies sent.
ACCESS      read-only
STATUS      mandatory
::= { echo 2 }

```

-- The Buffer Group

```

bufferSize OBJECT-TYPE
SYNTAX      INTEGER
-- DEFINITION The size of a buffer including the header.
ACCESS      read-only
STATUS      mandatory
::= { buffer 1 }

```

bufferAvail OBJECT-TYPE

SYNTAX INTEGER

-- DEFINITION The total number of buffers initialized.

ACCESS read-only

STATUS mandatory

::= { buffer 2 }

bufferDrops OBJECT-TYPE

SYNTAX Counter

-- DEFINITION The total number of requests for a buffer that were
-- denied.

ACCESS read-only

STATUS mandatory

::= { buffer 3 }

bufferTypeTable OBJECT-TYPE

SYNTAX SEQUENCE OF BufferTypeEntry

-- DEFINITION The information about each buffer type.

ACCESS read-only

STATUS mandatory

::= { buffer 4 }

bufferTypeEntry OBJECT-TYPE

SYNTAX BufferTypeEntry

-- DEFINITION The information about a buffer of this type.

ACCESS read-only

STATUS mandatory

::= { bufferTypeTable 1 }

```
bufferTypeEntry ::= SEQUENCE {  
    bufferTypeIndex INTEGER,  
    bufferType      INTEGER,  
    bufferTypeDescr OCTET STRING,  
    bufferTypeCount INTEGER  
}
```

```

bufferTypeIndex OBJECT-TYPE
SYNTAX          INTEGER
-- DEFINITION   The internal index of this type of buffer; This
--              value ranges from 1 to the total number of buffer
--              types defined on this entity.
ACCESS          read-only
STATUS          mandatory
::= { bufferTypeEntry 1 }

bufferType OBJECT-TYPE
SYNTAX          INTEGER {
                    other(1),
                    free(2),
                    localtalk(3),
                    ethernet(4),
                    arp(5),
                    data(6),           -- general data
                    erbf(7),          -- ethernet receive
                    etbf(8),          -- ethernet transmit complete
                    malloc(9),
                    tkbf(10),         -- token ring receive packet
                    token(11)         -- token ring packet
                }
-- DEFINITION   The type of this type of buffer.
ACCESS          read-only
STATUS          mandatory
::= { bufferTypeEntry 2 }

bufferTypeDescr OBJECT-TYPE
SYNTAX          OCTET STRING
-- DEFINITION   A printable ASCII text description of this type of buffer.
ACCESS          read-only
STATUS          mandatory
::= { bufferTypeEntry 3 }

```

```
bufferTypeCount OBJECT-TYPE
SYNTAX          INTEGER
-- DEFINITION   The number of buffers of this type currently allocated.
ACCESS          read-only
STATUS          mandatory
::= { bufferTypeEntry 4 }

END
```

C

C

C

Index

Symbols

(pound sign) 6-13
./DESKTOP file 1-47, 7-14
/etc/atalk.local file 1-39, 4-24
/etc/atalkatab file 1-39, 4-23
/etc/exports file 7-3, 7-4
/etc/group file 7-5, 7-7
/etc/hosts file 1-20, 6-13, 7-5
/etc/passwd file 7-5, 7-6
/etc/printcap file 1-43, 6-2, 6-12
/etc/services file 4-23

Numerics

10Base2 2-8
10Base5 2-7
10BaseT 2-9

A

A/UX 1-50, 6-2
About GatorKeeper command 1-8
access restrictions 7-4
acknowledgment control packet 4-5
active discovery 4-11
Add Server command 1-26
Add Server dialog box 1-26

Additional TCP/IP MacIP Parameters
 dialog box 1-32
address assignment 4-5
Address Mapping Table (AMT) 3-18
address resolution (AppleTalk)
 DDP 3-19
 NBP 3-18
Address Resolution Protocol (ARP) 3-7
Apple menu 1-6
AppleDouble 1-47, 1-51, 7-11
AppleShare 7-1, 7-10
AppleShare Server Name field 1-46
AppleShare Volume Name field 1-47
AppleShare-to-NFS dialog box 1-46
AppleSingle 1-47, 7-11
AppleTalk 4-5
AppleTalk Address Resolution Protocol
 (AARP) 3-18
AppleTalk ARP Style field 1-32
AppleTalk broadcast address 4-5
AppleTalk Filter dialog box 1-37
AppleTalk Phase 1 4-13
AppleTalk Phase 2 4-13
AppleTalk routing
 default configuration 1-3
AppleTalk Routing dialog box 1-33
AppleTalk tunnel 1-37, 1-40, 4-15
AppleTalk Tunnel dialog box 1-40

- AppleTalk zones 4-3
- Apply Authentication dialog box 1-25
- area (DECnet) 1-42, 5-1
- area number (DECnet) 5-1
- arp -a command 3-7
- ARP cache 3-7
- atalkad 1-39, 4-23
- atalkatab 4-23

B

- backbone network 2-3
- bandwidth 2-4
- Berkeley Software Distribution 6-1
- binary numbering system 2-11
- bridge 2-3
- bridgenet 1-39
- bridgenode 1-39
- broadcast address 3-6
- Broadcast Address field 1-28
- broadcast message 3-6
- by Icon command 1-23
- by Name command 1-23
- byte-range locking 7-16

C

- Can't Find status 1-14
- Change Password command 1-21
- Change Password dialog box 1-21
- Cheapernet 2-8
- checksum 4-2
- Checksum packets field 1-49

- chmod command 7-8
- Chooser command 1-8
- Cleanup View command 1-20
- Clear command 1-12
- clearing 1-12
- client 2-3
- client node address 4-5
- Close command 1-9
- coaxial cable 2-4
- Columbia AppleTalk Package 4-24
- command
 - About GatorKeeper 1-8
 - Add Server 1-26
 - by Icon 1-23
 - by Name 1-23
 - Change Password 1-21
 - Chooser 1-8
 - Cleanup View 1-20
 - Clear 1-12
 - Close 1-9, 1-11
 - Diagnostics 1-15
 - GatorBoxes 1-12
 - Info 1-17
 - keyboard equivalents 1-7
 - Lookup in Zone 1-24
 - New 1-8
 - Open 1-9
 - Paste 1-11
 - Print 1-10
 - Quit 1-10
 - Reload Software 1-22
 - Rename GatorBox 1-20
 - Restart GatorBoxes 1-22
 - Save 1-9

- Save Info as TEXT File 1-9
- Select All 1-12
- Servers 1-13
- Statistics 1-18
- Status 1-14
- Undo 1-11
- configuration information
 - printing 1-10
 - saving 1-9
- Configuration Options window 1-27
- connection attempts 6-7
- Convert TEXT files field 1-48
- Copy command 1-11
- cost 5-3
- crash information 1-12, 1-17
 - printing 1-10
 - saving 1-10

D

- data fork 7-11
- datagram 2-12
- Datagram Delivery Protocol (DDP)
 - 3-16, 4-1
- DDP 3-16
- DDP header
 - long 4-2
 - short 4-1
- DDP-style address resolution 3-19
- decimal numbering system 2-11
- DECnet 5-1
- DECnet Configuration dialog box 1-42
- DECnet router
 - level 1 5-2

- level 2 5-2
- default gateway address 1-29
- default zone 4-11
- delimiter character 1-51
- device (NBP) filtering 4-17
- device name filtering 4-21
- diagnostic information
 - printing 1-10
 - saving 1-10
- Diagnostic Messages window 1-15
- Diagnostics command 1-15
- dialog box
 - Add Server 1-26
 - Add Servers 1-26
 - Additional TCP/IP MacIP Parameters 1-32
 - AppleShare-to-NFS 1-46
 - AppleTalk Filter 1-37
 - AppleTalk Routing 1-33
 - AppleTalk Tunnel 1-40
 - Apply Authentication 1-25
 - Change Password 1-21
 - KIP Options 1-38
 - MacIP Options 1-30
 - NFS Mount Points 1-47
 - Printer Configuration 1-43
 - Reload Software 1-22
 - Rename GatorBox 1-20
 - Restart GatorBox 1-22
 - Select Zone 1-24
 - Server List 1-13
 - TCP/IP Configuration 1-28
 - User/Group Information 1-48
 - Zone List 1-41

- directory structure 7-2
- displaying 1-17
- Domain Name 1-25
- Domain Name field 1-48
- dynamic address assignment (MacIP)
 3-16
- dynamic MacIP addresses 1-31

E

- Edit menu 1-6
- encapsulation 3-16
- enquiry control packet 4-5
- Ethernet hardware address 1-17
- EtherTalk Link Access Protocol (ELAP)
 4-1
- EtherTalk Network Number field 1-35
- EtherTalk Zone Name field 1-35
- export list 7-3

F

- Farallon Star Controller 6-9
- file access security 7-7
- file creation times/dates 7-14
- File menu 1-6
- file name mapping 1-50
- file server 2-2, 7-1
- File Server Address field 1-32
- File Transfer Protocol (FTP) 3-2
- filtering 1-36
 - laser 1-37
 - network 1-37
 - stay-in-zone 1-37

- tilde 1-37

- firmware release level 1-17
- Forward Request 4-8
- FwdReq 4-8

G

- gateway 2-4
- GatorBox
 - GatorBox CS 1-1
 - GatorMIM CS 1-1
 - original 1-1
- GatorBoxes command 1-12
- GatorBoxes window 1-5, 1-12
- GatorDatabase 1-4
- GatorDefaults 1-4
- GatorInstaller 1-4
- GatorKeeper 1-4
- GatorMIM CS 1-1
- GatorPrint CS 1-3
- GatorShare 7-1, 7-13
- GatorShare CS 1-3
- GatorShare Servers window 1-45
- GatorSystem CS 1-2
- Get Privileges function 7-10
- gid 7-10
- group file 1-49
- Group ID 7-10
- group security 7-7

H

- hardware address 3-3

- Hayes InterBridge 3-17
- Hello message 5-5
- HELLO timer 5-3
- Hello timer 1-42
- hexadecimal numbering system 2-12
- hop count 4-2, 5-3
- host 2-2

I

- ifconfig command 3-6
- Info command 1-17
- Info window 1-17
- InterBridge 3-17
- international character mapping 6-10
- international character set 6-10
- international characters 1-44
- internet 2-1
- internet (IP) address 3-3
- Internet Control Message Protocol (ICMP) 3-2
- Internet Protocol (IP) 3-1
- IP address
 - class A 3-5
 - class B 3-5
 - class C 3-5
 - GatorBox 1-28
- IP Address field 1-28
- IP routing 3-13
- IP subnet 1-30
- IP subnetting 3-8
- IPGATEWAY device type 3-16
- IPTalk 4-22

- ISO 8859-1 1-44, 6-10

K

- Kinetics Internet Protocol 4-22
- KIP 1-38, 4-22
- KIP AppleTalk Network Number field 1-39
- KIP Options dialog box 1-38
- KIP-style forwarding 1-30

L

- LAN 2-1
- landscape layout 6-9
- laser filtering 4-20
- LaserJet 6-9
- LaserSpooler 6-9
- LaserWriter 6-9
- Link Access Protocol (LAP) 4-1
- local area network 2-1
- LocalTalk 2-5, 4-1
- LocalTalk connector boxes 2-5
- LocalTalk Link Access Protocol (LLAP) 4-1
- LocalTalk Network Number field 1-34
- LocalTalk Printer Name field 1-43
- LocalTalk Printer Type field 1-43
- LocalTalk Printer Zone field 1-44
- LocalTalk Zone Name field 1-34
- logical printer 6-6
- long DDP headers 4-2
- Lookup in Zone command 1-24

lpd 6-1
lpr 1-3, 6-1
lpr control file 6-1
lpr data file 6-1
lprclient utility 6-3

M

MacIP 1-30, 3-16
MacIP address assignment
 dynamic 3-16
 static 3-17
MacIP Options dialog box 1-30
MacTCP 3-16
mail server 2-2
Management Information Base 8-11
maximum file size 6-14
maximum segment size 5-3
memory allocation 1-19
memory partition 1-19
menu
 Apple 1-6
 Edit 1-6
 File 1-6
 Server Access 1-6
 Special 1-6
 View 1-6
 Windows 1-6
menu bar 1-7
minimum cost calculation 5-4
mount point 7-1, 7-3
mountd 7-5
Multi-Media Access Center 1-1

mynet 1-39
myzone 1-39

N

name,GatorBox 1-20
Name Binding Protocol (NBP) 4-2
name server 2-2
Name Server Address field 1-32
Namer utility 6-8
NBP Broadcast Request 4-8
NBPBrRq 4-8
NBP-style address resolution 3-18
NCSA Telnet 1-2, 3-2, 3-16
NetBridge 3-17
NetInfoManager application 6-13
netstat -i command 3-6
netstat -r command 3-15
network addresses 2-10
Network File System 7-1
network filter 4-16
network filtering 1-36
Network Information System 7-5, 7-8
network media 2-4
network range 4-11
network segment (of IP address) 3-4
New command 1-8
NeXT 6-3
NextBridge 4-4
NFS Mount Point field 1-47
NFS Mount Points dialog box 1-47
nfsd 7-5

- node 2-2
- node (DECnet) 5-1
- node number (DECnet) 5-1
- node segment (of IP address) 3-4
- noise resistance 2-4
- nonseed router 4-11
- nonseed routing 4-10

O

- Open command 1-9
- owner security 7-7

P

- packets 2-12
- page format 1-10
- PAP (Printer Access Protocol) 6-1, 6-4
- PAP connection timer 6-4
- paper size 1-10, 1-44
- parameter random access memory (PRAM) 4-6
- passive discovery 4-11
- password 1-21
- password aging 7-10
- password file 1-49
- Paste command 1-11
- PATHWORKS 5-2
- PC AppleShare 7-11
- pcnfsd 1-26, 1-49, 7-9
- Phase 1/Phase 2 transition 4-14
- PhoneNET 2-6
- PhoneNET terminator 2-6

- physical printer 6-6
- ping 3-4
- portmapper 7-4
- portrait layout 6-9
- PostScript 6-5
- PostScript conversion 6-5
- PRAM (parameter random access memory) 4-6
- Prefix used for AppleDouble resource file field 1-51
- print client 6-1, 6-4
- Print command 1-10
- print server 2-2
- printcap file 6-2
- Printer Access Protocol (PAP) 6-1, 6-4
- Printer Configuration dialog box 1-43
- Printer LPR Name field 1-43
- printer name 6-8
- printer server 6-1, 6-4
- printer type 6-8
- PrintManager tool 6-13
- protocols 2-9
- proxy ARP 3-7
- ps -ax command 7-5
- ps -ef command 7-5

Q

- Quit command 1-10

R

- Read size field 1-50
- reduction ratio 1-10
- Reload Software command 1-22
- Reload Software dialog box 1-22
- Rename GatorBox command 1-20
- Rename GatorBox dialog box 1-20
- repeater 2-2, 2-3
- resource fork 1-51, 7-10
- Restart GatorBox dialog box 1-22
- Restart GatorBoxes command 1-22
- Restrict NBP Lookups to LocalTalk field 1-33
- Retry count field 1-50
- Reverse Address Resolution Protocol (RARP) 3-8
- RFC 919 3-6
- RFC 922 3-6
- RIP 1-29
- route add command 3-16
- router 2-3
- Routing Information Protocol (RIP) 3-15
- routing messages 5-5
- routing table 2-4, 3-13, 4-3
- routing table (DECnet) 5-3
- Routing Table Maintenance Protocol (RTMP) 4-2
- routing timer 1-42
- rpcinfo -p command 7-5
- RTMP packet 4-2
- Running status 1-14

S

- Save command 1-9
- Save Info as TEXT File command 1-9
- seed router 4-11
- seed routing 4-10
- Select All command 1-12
- Select Zone dialog box 1-24
- serial number 1-17
- Server Access menu 1-6
- Server List 1-26
- Server List dialog box 1-13
- server node address 4-5
- Servers command 1-13
- shadow passwords 7-10
- shielded cable 2-5
- Shiva NetBridge 3-17
- short DDP headers 4-1
- show alap 8-7
- show appletalk arp 8-4, 8-5, 8-6
- show appletalk routes 8-3
- show appletalk zones 8-4
- show crash 8-9
- show enet 8-8
- Show invisible files field 1-48
- show ip arp 8-3
- show ip routes 8-2
- show log 8-8
- show memory 8-9
- show share 8-7
- Show Volume(s) field 1-48
- showmount command 7-4

Silicon Graphics Iris 6-3
Simple Mail Transfer Protocol (SMTP) 3-3
Simple Network Management Protocol 8-10
SNMP 8-10
socket 4-1
soft seed router 4-12
soft seed routing 4-10
software release level 1-17
Special menu 1-6
spool directory 6-14
SRI Network Information Center 3-5
static address assignment (MacIP) 3-17
static MacIP addresses 1-31
Statistics command 1-18
Status command 1-14
stay-in-zone filtering 1-37, 4-18
subnet 1-31
subnet broadcast address 3-12
subnet mask 3-10
Subnet mask field 1-29, 1-31
subnetting 3-8
Sun-1 1-50
Sun-2 1-50
suspect route 4-2
syslog host address 1-29

T

T-connector 2-8
TCP/IP Configuration dialog box 1-28
TELNET 3-2

Telnet (NCSA) 3-2
TELNET shell 8-1
TELNET syntax 8-1
terminator 2-7
TeX 6-10
Text-to-PostScript filter 1-44
thick Ethernet 2-7
thin Ethernet 2-8
tickle packet 6-4
tilde (device name) filtering 4-21
Timeout field 1-50
tn3270 3-19
Token Ring 4-1
TRANSCRIPT 6-10
Transmission Control Protocol (TCP) 3-1
transmission speed 2-4
Trivial File Transfer Protocol (TFTP) 3-2
troff 6-10
tunnel, AppleTalk 1-37
twisted pair cable 2-5

U

UDP port range 1-40, 4-22
UDP time service 7-14
uid 7-10
ULTRIX 6-1
Unconfigured status 1-14
Undo command 1-11
unshielded cable 2-5
User Datagram Protocol (UDP) 3-2
User ID (uid) 7-10

User's Home Directory 7-10
User/Group Information dialog box
1-48

zone tables 4-4

V

View menu 1-6
volume 7-1
Volume Password field 1-47

W

WAN 2-1
wide area network 2-1
window
 Configuration Options 1-27
 Diagnostic Messages 1-15
 GatorBoxes 1-12
 GatorShare Servers 1-45
 Info 1-17
Windows menu 1-6
Write size field 1-50

Y

Yellow Pages 1-25
Yellow Pages field 1-48
ypmatch command 7-9

Z

Zone Information Protocol (ZIP) 4-4
zone list 4-11
Zone List dialog box 1-41
zone name 1-34

Reader Reply Card

Cayman is interested in learning how we can improve our documentation and customer support. Please take a moment to complete this postage-paid survey. Your comments are greatly appreciated.

1. How do you use this manual: (Check one or more)

- | | |
|--|--|
| ☐ To get an overview of the product | ☐ To get out of trouble |
| ☐ To learn a task | ☐ Other _____ |
| ☐ To look up a fact | _____ |

2. How often do you use this:

Manual?

Product?

- | | | | | | |
|--------------------------------|---------------------------------|---------------------------------------|--------------------------------|---------------------------------|---------------------------------------|
| ☐ Daily | ☐ Weekly | ☐ Infrequently | ☐ Daily | ☐ Weekly | ☐ Infrequently |
|--------------------------------|---------------------------------|---------------------------------------|--------------------------------|---------------------------------|---------------------------------------|

3. Is the information accurate, easy to find, and easy to read?

- ☐ Yes ☐ No _____

4. Are the examples helpful and realistic?

- ☐ Yes ☐ No _____

5. Are the illustrations helpful, realistic, and easy to read?

- ☐ Yes ☐ No _____

6. Is the index complete and accurate?

- ☐ Yes ☐ No _____

7. Did you find any errors? (Please give page numbers) _____

8. Did you notice any omissions? (Please give page numbers)

9. Do you have any general comments or suggestions? _____



Tape here

May we call you if we have questions about your comments? Yes ☐ No ☐

Telephone _____

City/State/ZIP _____

Address: _____

Company: _____

Name: _____

Respondent information (optional):

Fold here



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 6480 BOSTON, MA

POSTAGE WILL BE PAID BY ADDRESSEE

CAYMAN SYSTEMS, INC.
UNIVERSITY PARK AT MIT
26 LANDSDOWNE STREET
CAMBRIDGE, MA 02139-9732



Reader Reply Card

Cayman is interested in learning how we can improve our documentation and customer support. Please take a moment to complete this postage-paid survey. Your comments are greatly appreciated.

1. How do you use this manual: (Check one or more)

- | | |
|--|--|
| ☐ To get an overview of the product | ☐ To get out of trouble |
| ☐ To learn a task | ☐ Other _____ |
| ☐ To look up a fact | _____ |

2. How often do you use this:

Manual?

Product?

- | | | | | | |
|--------------------------------|---------------------------------|---------------------------------------|--------------------------------|---------------------------------|---------------------------------------|
| ☐ Daily | ☐ Weekly | ☐ Infrequently | ☐ Daily | ☐ Weekly | ☐ Infrequently |
|--------------------------------|---------------------------------|---------------------------------------|--------------------------------|---------------------------------|---------------------------------------|

3. Is the information accurate, easy to find, and easy to read?

- ☐ Yes ☐ No _____

4. Are the examples helpful and realistic?

- ☐ Yes ☐ No _____

5. Are the illustrations helpful, realistic, and easy to read?

- ☐ Yes ☐ No _____

6. Is the index complete and accurate?

- ☐ Yes ☐ No _____

7. Did you find any errors? (Please give page numbers) _____

8. Did you notice any omissions? (Please give page numbers)

9. Do you have any general comments or suggestions? _____



Colophon

Documentation

This manual was created using Microsoft Word and FrameMaker on a Macintosh SE/30 and a Macintosh IIfx. Art was produced using Adobe Illustrator and Claris MacPaint. Proof pages were produced using an Apple Personal LaserWriter. Final pages were produced using a LaserMax 1000ks Personal Typesetter. Typefaces for this manual are Adobe Stone Sans, Garamond, Courier, and Universal News.

Writing/Artwork: Michael McCoy

Editing: Nancy Rawlings

Product Development

Hardware Development: Ted Bilodeau, Mike Eaton, Marc Harrison, James Pinkerton, Fan-Chia Tao

Software Development: Pong Choa, Paul G. Fox, Art Mellor, Brad Parker, Ken Siegel, John Stephens, Maria Yunis

Manufacturing Process Development: Ina Bresner, Paul Couto, Charles Crosby, Joe Geary, Bill Kirtley, Joe Salesi, Mark Schlepphorst

Quality Assurance: Gil Côté, Karen Houldin, Pat McLaughlin, Colin A. Steele

Technical Services: Cindy Brown, Christopher Evans, Glen B. Glater, Michael Haag, Throop Wilder

Product Management: Susan Gladstone







20







Cayman Systems
26 Landsdowne Street
Cambridge, MA 02139
(617) 494-1999

GatorBox Reference
12099-1 Rev A